

Reconocimiento de gestos de la mano aplicado a una interfaz para ambientes de aprendizaje

Ramón Zatarain-Cabada, María Lucia Barrón-Estrada,
Brandon Antonio Cárdenas Sainz

Instituto Tecnológico de Culiacán, Culiacán, Sinaloa, México
{rzatarain, lbarron, brandon_cardenas}@itculiacan.edu.mx

Resumen. En este artículo presentamos un prototipo de una aplicación web interactiva 3D, la cual combina la interfaz de Google Blockly con una interfaz de interacción con implementación de reconocimiento de gestos de manos humanas. La aplicación consiste en que el usuario final tenga las posibilidades de programar sus propios gestos en 3D, asociarlos a eventos e implementarlos dentro de un entorno gráfico, mientras se utiliza programación basada en bloques para la creación de aplicaciones. Todo con el fin de desarrollar el pensamiento computacional del individuo y que este se familiarice con conceptos de interfaces humano-máquina naturales, programación algorítmica y pensamiento espacial. Aplicando evaluaciones de diseño centrado en el humano, se demuestra que las interfaces naturales y gráficas presentadas en este proyecto cumplen con los principios básicos de interacción natural y se tienen expectativas que indican que nuestro enfoque impacta de forma positiva el proceso de aprendizaje, aumentando la motivación y disfrute.

Palabras clave: interacción humano-máquina, interfaces naturales de usuario, pensamiento computacional, reconocimiento de gestos.

Recognition of Hand Gestures Applied to an Interface for Learning Environments

Abstract. This paper presents a prototype of an interactive 3D web application, which combines the Google Blockly interface with an interaction interface using an implementation of recognition of human-hand gestures. The application allows the end-user to be able to program their own 3D gestures, associate them to events and implement them within a graphical environment while using block-based programming for the creation of applications. The main goal is to develop the computational thinking of the individual and that becomes familiar with concepts of natural human machine interfaces, algorithmic programming, and spatial thinking. By applying human-centered design evaluations, it is demonstrated that the natural and graphic interfaces presented in this project comply with the basic principles of natural interaction and have expectations that indicate that our approach positively affects the learning process, increasing motivation and enjoyment.

Keywords: human-machine interaction, natural user interfaces, computational thinking, gesture recognition

1. Introducción

El sector educativo busca más que nunca en la actualidad, diversas formas para el desarrollo del pensamiento computacional. Esta es una serie de habilidades que integran la resolución de problemas, el diseño de sistemas y algoritmos, así como el

entendimiento del comportamiento humano, aplicando para ello conceptos estructurales de la computación. Este proceso se enfoca en el reconocimiento de aspectos de la informática en el mundo que nos rodea, y aplica herramientas y técnicas de la informática para comprender y razonar sobre los sistemas y procesos tanto naturales como artificiales [1].

Usualmente, cuando se habla de informática y de la computación, o de temas como la programación e interacción con una computadora, nos imaginamos a alguien sentado sobre una mesa, escribiendo sobre un teclado y moviéndose en pantalla por medio de un ratón o touchpad. Esta forma de interactuar con las computadoras no ha cambiado significativamente desde 1960 [2], época en la que estos periféricos fueron inventados. Actualmente existen diversos dispositivos en el mercado tales como tabletas y teléfonos inteligentes, en donde se presentan interfaces más innovadoras: pantallas con capacidad táctil y el uso de gestos que representan una forma más natural de interacción entre el humano y la máquina. También se han realizado diversas implementaciones de reconocimiento de gestos en dispositivos con sensores o cámaras con la finalidad de ofrecer ciertas simplicidades de uso [3].

El reconocimiento de gestos es un tema en las ciencias de la computación y de la tecnología del lenguaje con el objetivo de interpretar gestos humanos a través de algoritmos matemáticos. Los gestos pueden ser cualquier movimiento corporal o estado, pero comúnmente se originan a partir de un rostro o de las manos. El reconocimiento de gestos puede ser visto como la manera en que las computadoras interpretan el lenguaje corporal humano, considerándose como una interfaz natural humano-computadora.

Esto nos lleva a los conceptos de interacción humano-máquina (HMI) y las interfaces naturales de usuario (NUI), donde esta última define que los humanos se pueden comunicar con la máquina e interactuar naturalmente sin dispositivos mecánicos [2]. Utilizando el concepto de reconocimiento de gestos, es posible usar los dedos en un espacio libre para relacionar el movimiento del cursor con los movimientos del usuario.

Actualmente, la presencia de interfaces humano-máquina más naturales se ha vuelto una necesidad, ya que estas pueden proporcionar nuevas capacidades y facilidad de uso en diversos dispositivos computacionales, ya sean computadoras, teléfonos inteligentes, tabletas, máquinas del sector industrial, etc. Estas interfaces ofrecen al usuario una mejora en el tiempo de aprendizaje, la velocidad de desempeño, a la tasa de error de uso y la forma en que un usuario opera el software de un dispositivo y la satisfacción que se experimenta en el uso [4].

Por otra parte, el desarrollo del pensamiento computacional y de que un individuo sea capaz de programar sigue siendo hoy en día un problema abordado por diversas investigaciones. Desde la creación de herramientas digitales llamadas e-learning [5] para facilitar el aprendizaje, sistemas o mecanismos de asistencia de acuerdo al rendimiento del individuo (sistemas tutores inteligentes STI) [6], la comprensión de la conducta humana y de sus estados emocionales y como estos afectan su rendimiento a la hora de aprender (sistemas tutores afectivos STA) [7], son algunos de los enfoques que ofrecen diversas soluciones. Uno de estos enfoques es cómo el usuario llega a interactuar con un dispositivo de cómputo, herramientas en las cuales se realiza el ejercicio cognitivo computacional.

Como se mencionaba anteriormente, algunos estudios indican que la utilización de interfaces humano-máquina con un enfoque a la interacción natural ayuda a la visualización, la comprensión y por ende el aprendizaje del usuario [8] por lo que puede ser de gran ayuda para el sector educativo.

Por ello, el objetivo de esta investigación es la exploración de nuevas formas de interacción implementadas en entornos de aprendizaje, así como el impacto sobre el proceso de aprendizaje, aumentando la motivación, el disfrute del estudio y el desarrollo del pensamiento computacional.

Este artículo está organizado de la siguiente forma: la Sección 2 describe los trabajos relacionados. La Sección 3 presenta la estructura y funcionamiento de la herramienta. En la Sección 4 se muestra una sesión de trabajo. La Sección 5 contiene algunos de los experimentos realizados y los resultados obtenidos. Por último, en la Sección 6 se describen las conclusiones y trabajos futuros.

2. Trabajos relacionados

En esta sección se describen investigaciones y trabajos en áreas relacionadas con el campo del reconocimiento de gestos y el pensamiento computacional en la educación.

Una de las herramientas más importantes que se pueden mencionar es Scratch [9], una comunidad de aprendizaje creativo en línea, donde los usuarios pueden crear sus proyectos a partir de bloques gráficos, donde un bloque representa un elemento del lenguaje de la programación, como son: control, operadores, variables, funciones, etc. Otra herramienta importante es Alice [10], en donde principiantes pueden programar animaciones y juegos 3D utilizando modelos ya predefinidos a partir de una interfaz visual basada en la programación por bloques.

En el marco del uso de interfaces naturales, mediante el reconocimiento de gestos en el campo de la educación, los trabajos de investigación más importantes han estado dedicados principalmente a estudiar factores como la usabilidad y la motivación en prototipos educativos. S. Bowe, M. Antoniou, C. Garrett et al. [11] presentan el aprendizaje de niños y jóvenes utilizando dispositivos con pantalla táctil, donde se realiza una actividad con un juego de la Torre de Hanoi, dentro de una aplicación Android, para luego resolver uno real, concluyendo que el conocimiento y habilidades aprendidas en una tarea desde un dispositivo móvil pueden ser transferidas en el mundo real. De acuerdo con el trabajo de M. Ebner et al. [12] el uso de interfaces innovadoras como el controlador Leap Motion (LMC), mejora significativamente el proceso de aprendizaje dentro de aplicaciones que educan mediante el juego. Estas mejoran la motivación, el enganche y la diversión de los estudiantes mientras juegan aprendiendo. G. Zhu, S. Cai, y. Ma et al. [13] usan LMC para analizar la efectividad para la rehabilitación motriz de niños con autismo a partir de una serie de ejercicios en aplicaciones para el sector escolar. V. Tran, J. Lee, D. Kim et al. [14] en colaboración con The LEGO Group, LEGO System A/S Denmark usaron los dispositivos LMC y Kinect para presentar en VR una actividad interactiva que permitía al usuario diseñar y construir edificaciones con piezas LEGO.

Hasta el momento, no existe un sistema integral, con el enfoque del cómo el usuario interactúa con un dispositivo de cómputo y con la premisa de que el usuario aprenda y pueda dominar los conceptos que componen el pensamiento computacional, a partir de la programación e implementación de las mismas interfaces interactivas dentro de un entorno de aprendizaje.

La contribución principal de este proyecto es presentar un prototipo de un entorno de aprendizaje que englobe diferentes interfaces de interactividad natural, así como el uso de interfaces gráficas de programación mediante bloques y de entornos gráficos interactivos, con el fin de ofrecer al usuario final la capacidad de programar el modo de interacción entre un dispositivo de cómputo.

3. Interfaz de programación visual y reconocimiento de gestos

Para la investigación de este proyecto, se ha creado un prototipo de una aplicación web, la cual consiste en que el usuario final tenga las posibilidades de programar sus propios gestos en 3D, asociarlos a eventos e implementarlos dentro de un entorno gráfico, mientras se utiliza programación basada en bloques para la creación de aplicaciones. El principal objetivo es desarrollar el pensamiento computacional del individuo y que este se familiarice con conceptos de interfaces humano-máquina naturales, programación algorítmica y pensamiento espacial, etc. La figura 1 nos muestra un diagrama de los componentes del prototipo y las librerías utilizadas.

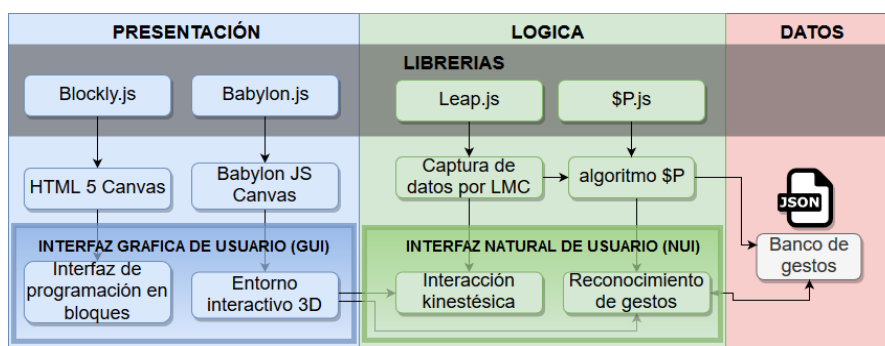


Fig. 1. Diagrama de componentes del prototipo interactivo.

El prototipo posee un entorno de interacción 3D con un sistema de físicas y tiene como propósito que dentro de un escenario virtual el jugador cree e interactúe con elementos 3D. La aplicación muestra al usuario una interfaz de programación visual para que genere elementos dentro de un escenario, creando cuerpos geométricos tridimensionales, los cuales pueden poseer una masa y ser afectados por simulación de efectos de gravedad, así como aplicación de fuerzas, velocidades e impulsos tanto angulares como lineales. Además, el usuario final también tiene la posibilidad de programar eventos, acciones y la forma en que sus manos interactúan con la escena 3D y sus objetos.

El entorno interactivo está ubicado en un escenario 3D (ver figura 7 en sección 4.2). Su presentación más básica consiste en un plano 3D que funciona como terreno, sin embargo, se tiene la posibilidad de cargar diferentes escenarios. Este incluye un avatar, un personaje caricaturesco redondo que posee un par de manos flotantes, las cuales son utilizadas para representar los gestos de manos. El avatar también representa la posición en la que se encuentra el jugador y ofrece un punto de visión de las manos dentro del entorno. El usuario a partir de bloques, programa la creación de elementos dentro del entorno gráfico y las funciones de interacción que implementan el reconocimiento de gestos.

3.1. Reconocimiento de gestos con Leap Motion

Leap Motion es un periférico USB principalmente diseñado para la detección de la posición de dedos y gestos de una mano para software de aplicaciones interactivas. Posee una API de desarrollo la cual provee acceso a un modelo de las características

vectoriales del esqueleto de una mano detectada y está diseñado para ser situado en un escritorio, mirando verticalmente. El campo de visión tiene una forma cónica, la cual se extiende entre 2.5 a 60 cm por encima del dispositivo. El Controlador de Leap Motion (LMC) posee dos cámaras infrarrojas monocromáticas y tres LEDs infrarrojos. Esto permite al software del controlador generar un patrón 3D a partir de los datos 2D obtenidos por ambas cámaras (ver figura 2).

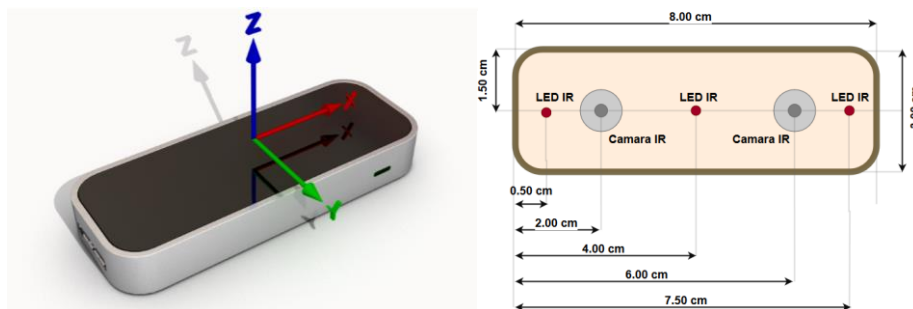


Fig. 2. Dimensiones del controlador Leap Motion.

3.2. Reconocimiento de gestos en entornos 2D y 3D y aprendizaje máquina

Este proyecto se enfoca principalmente en realizar reconocimiento de gestos interpretando a este como el trayecto realizado por las puntas de los dedos de una mano, simulando la información que se recibe al utilizar algún dispositivo táctil, pero aplicado a un entorno 3D. Esto último implica que debe existir alguna forma de registrar la trayectoria de una mano no solo en una perspectiva 2D, sino también 3D.

Para el problema de localización de la posición de una mano de acuerdo a la profundidad de un entorno este proyecto se ha enfocado en dar a conocer 2 propuestas: utilizar un algoritmo que pueda reconocer gestos realizados tanto en 2D como 3D y determinar campos de profundidad de un entorno a partir de áreas de interacción.

“SP” es un algoritmo que tiene como propósito ofrecer una solución rápida de implementación de reconocimiento de gestos a proyectos que requieran interacciones humana-máquina naturales [15]. Este representa a un gesto como una nube de puntos que, al unirlos entre sí, forman un trazo o trayectoria geométrica que después es rotada, escalada y normalizada, para que sea comparada con una base de datos que almacene gestos (ver figura 3). En términos de aprendizaje máquina, SP es un clasificador basado en KNN con una función de distancia euclidiana 2D, es decir, un igualador de plantillas geométricas.

Para el reconocimiento de gestos se ha adaptado el algoritmo SP para su uso en 3D, representando a un gesto como el desplazamiento de las puntas de los dedos de una mano, formando un trazo o una serie de puntos en un plano. A partir del dispositivo LMC, se obtienen características de las manos humanas que se encuentren dentro de su campo de visión. La información es almacenada en forma de series de fotogramas o frames, parecidos a tomar video, pero en su lugar se registran datos vectoriales de la posición, orientación, grados de inclinación de cada parte que compone a la mano humana, desde los huesos de la muñeca, las articulaciones y huesos de los dedos.

Esta información es procesada y registrada en un archivo en formato JSON, el cual contiene los datos vectoriales de cada frame, con su respectivo tiempo de captura. Esto permite que se creen animaciones de acuerdo a la duración del movimiento o gesto.

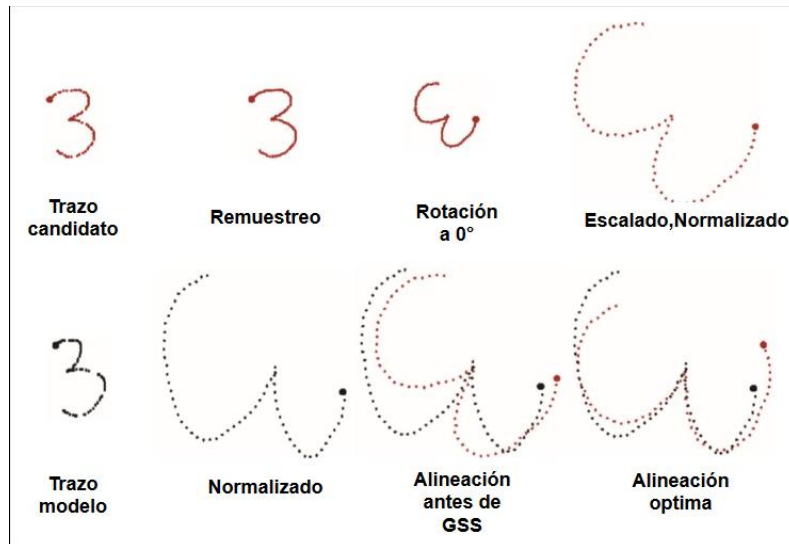


Fig. 3. Pasos del proceso de reconocimiento en \$P recognizer. GSS significa Golden Section Search [15].



Fig. 4. Representación de las áreas de interacción con respecto a la profundidad entre el usuario y el LMC.

Para el reconocimiento de los gestos 3D, se ha optado por separar el gesto realizado por las puntas de los dedos de una mano en sus diferentes perspectivas de acuerdo a los planos ortogonales XY y XZ. Sin embargo, implementar un sistema de reconocimiento de gestos dentro de entornos 3D implica también reconocimiento de la posición de la mano dentro de la misma en coordenadas 3D. Por lo que es necesario delimitar en qué momento un gesto es usado. Se definen 3 estados dentro del área de visión del sensor, un estado donde las manos son representadas, pero la interactividad y gestos están desactivados (PASIVO), el estado en la que la interactividad esta activada, pero el reconocimiento de gestos se mantiene desactivado (ACTIVO) y el estado cuando se

toca el límite frontal del LMC para activar reconocimiento de gestos (LIMITE) (ver figura 4).

3.3. Interactividad kinestésica en entornos 3D

Si bien, el reconocimiento de gestos forma parte de una interfaz humana-máquina natural, este no proporciona capacidades de interacción kinestésicas dentro del entorno. Es decir, que el usuario pueda interactuar con los elementos que se encuentre dentro de un escenario virtual de la misma manera que se realiza en la vida real, como levantar un objeto con sus manos, moverlo de un lugar a otro, lanzarlo, etc.

Esta es una característica que un sistema de reconocimiento de gestos no puede atender, puesto que este no utiliza información relacionada a por ejemplo verificar, si las manos dentro de un entorno virtual colisionan o se encuentran cerca de un objeto interactivo. Por lo cual, crear un componente de interacción para esta clase de entornos también forma parte de la interfaz de este proyecto.

Para que el usuario final dentro de la aplicación pueda interactuar directamente con sus manos, se utiliza una representación tridimensional de las manos, que consiste de un modelo esquelético para el entorno gráfico 3D. Además, la interfaz cuenta con un componente dedicado a la interacción mediante eventos a partir de ciertas posturas y acciones de las manos.

Este componente primero utiliza la información recibida por el sensor para evaluar la distancia entre el dedo pulgar y el dedo índice (picking). Después, evalúa que tanto la palma de la mano está cerrada o abierta (grabbing) y si el dedo índice está apuntando a un objeto a distancia (pointing) mientras se determina si la posición de la mano virtual se encuentra cerca de un elemento interactivo con el que se pueda realizar alguna acción. Esto ofrece formas en la que el usuario puede interactuar con los elementos dentro del entorno y también están disponibles para ser utilizados para programar eventos personalizados.

3.4. Interfaz de programación visual Blockly

El usuario final utiliza bloques que representan una serie de comandos para la creación de código, lo cual permite que el ejercicio de codificación se concentre más en la lógica del problema, lo que apoya el proceso de aprendizaje de los principiantes.

En la aplicación, Blockly [16]. Se encarga de proporcionar bloques que representan constructores de figuras geométricas tridimensionales, funciones básicas de personalización, configuración de posición, así como funciones orientadas al sistema de simulación de físicas, donde se aplican propiedades como masa, velocidad, impulsos, fuerza, gravedad, etc. Esto con el fin de presentar las funciones necesarias para la creación de objetos dentro del escenario. Además, se presentan bloques de ciclos, lógica y variables que tiene por defecto Blockly. (ver más adelante figura 6 en sección 4.2).

También se proporciona bloques relacionados a la programación de eventos de interacción y de reconocimiento de gestos, donde el usuario final puede programar en qué momento o bajo que circunstancia ocurre alguna acción relacionada al interactuar con los elementos del escenario. Esto significa que, a partir de la programación de eventos interactivos mediante gestos, se puede cambiar de color un objeto, cambiar de tamaño, moverlo, rotarlo, etc. También se pueden programar eventos de interacción kinestésicos, donde al sostener un objeto y después soltarlo, que el objeto sea lanzado,

o programar eventos de áreas de interacción, donde si una mano se encuentra fuera o dentro del campo de visión del sensor se controle la gravedad de toda la escena, etc.

4. Uso de la aplicación web

El usuario tiene la posibilidad de utilizar 2 secciones dentro de la aplicación. La primera llamada CREA está enfocada a la importación o exportación de un corpus de gestos. La segunda llamada JUEGA que consiste en el entorno de interacción que utiliza una interfaz visual basada en bloques para la programación de escenarios interactivos y que utiliza la carga de corpus de gestos creados por el propio usuario para la implementación de reconocimiento de gestos dentro de la interfaz de interacción.

4.1. CREA tus propios gestos

CREA consiste en la creación de un corpus de reconocimiento de gestos, donde a partir de la técnica de *programación por ejemplos* [17], se crea un archivo JSON que representa el corpus de gestos con su respectiva etiquetación, la cual puede ser cargado en los entornos del apartado JUEGA. Para crear gestos, el usuario usa sus manos sobre el sensor LMC, mientras que el software automáticamente procede a capturar datos y etiquetar por tiempo (en milisegundos) los fotogramas (frames) de lo que dura el gesto. La figura 5 muestra la interfaz del entorno de desarrollo CREA.

CREA permite al usuario visualizar los gestos producidos por medio de un reproductor que muestra una representación animada del gesto en 3D, y que incluye la visualización de las trayectorias realizadas por las puntas de sus dedos en perspectivas XY y XZ. El visualizador también apoya al usuario con un verificador incorporado que mide la exactitud de los gestos, comparándolo con el contenido del corpus generado.

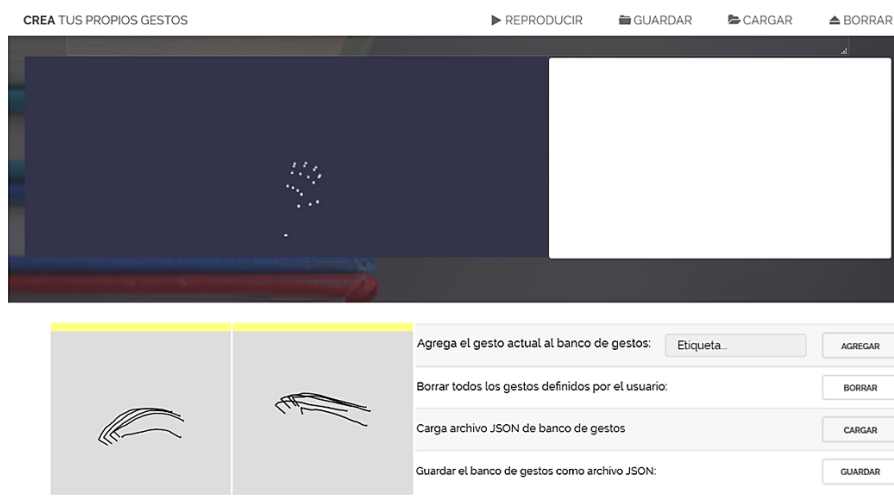


Fig. 5. Entorno de Desarrollo de CREA

Para crear un archivo JSON que represente el corpus de gestos, el usuario deberá de generar un gesto y después etiquetarlo con un nombre. Se puede repetir el registro del gesto y el proceso de etiquetación las veces que se desee. El archivo JSON resultante

se guarda en la computadora y este puede ser utilizado dentro del entorno JUEGA como parte de la interfaz de interacción, la cual implementa el reconocimiento de los gestos.

4.2. JUEGA y diviértete

JUEGA es el entorno interactivo que implementa una interfaz de interacción mediante reconocimiento de gestos y una interfaz visual orientada a la programación mediante bloques (ver figura 6).

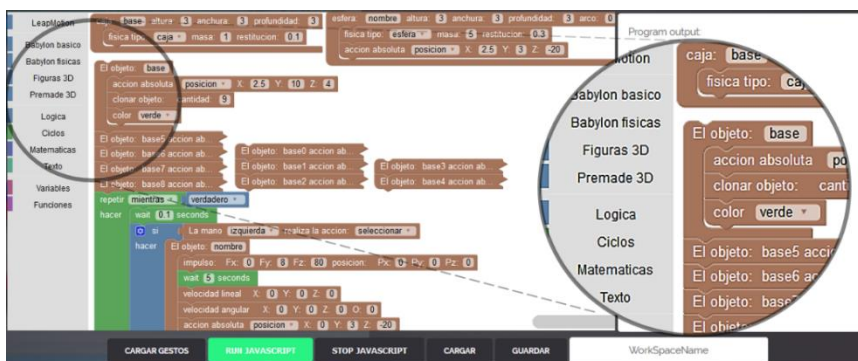


Fig. 6. Entorno de programación visual con Blockly.

A partir de los bloques, el usuario final construye elementos personalizados dentro del entorno, los cuales poseen físicas y pueden interactuar entre sí. También hay bloques para programación de la lógica y control de eventos que implementan la interfaz de interacción. Esto con el fin de que el usuario mediante gestos manipule e interactúe con lo que ocurre en el entorno 3D utilizando directamente sus manos de una forma más kinestésica, o bien utilizando el sistema de reconocimiento de gestos y relacionarlos a eventos o acciones en pantalla, a partir de la perspectiva de un avatar (ver figura 7).

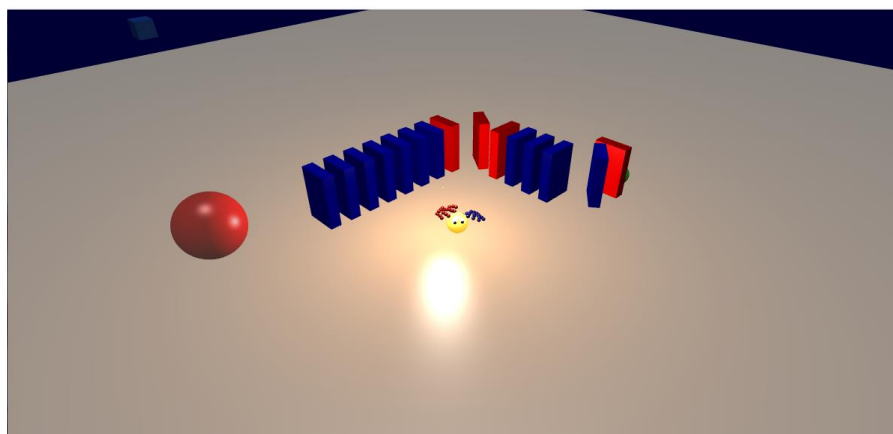


Fig. 7. Escena 3D con interacción kinestésica desde un avatar.

reconocimiento de gestos, por lo que CREA y JUEGA pueden ser utilizados en conjunto.

Respecto al ejercicio de programar un entorno interactivo, se puede dividir en 3 aspectos:

Creación de constructores de elemento 3D. Se utilizan bloques para la construcción de figuras geométricas con sus respectivas dimensiones. Se crea instancias de figuras 3D y se les asigna un nombre.

Personalización de elementos 3D y del entorno. Ya sea durante o después del proceso de construcción, se usan bloques que permiten personalizar el aspecto y las propiedades físicas de una figura. Permiten definir el comportamiento del objeto a partir del sistema simulador de físicas. Para que la gravedad simulada haga efecto, por ejemplo, se necesita que el objeto se le asigne una masa.

Creación del comportamiento lógico de entorno mediante eventos. Se realiza la programación de la lógica de eventos y las interacciones que pueden ocurrir por las acciones del usuario. Se programa a partir de bloques de secuencias y funciones lógicas, permite crear animaciones en el entorno, o programar eventos que utilicen la interfaz de interacción e implementar reconocimiento de gestos.

5. Evaluación de diseño centrado en el humano

El diseño centrado en el humano (DCH) tiene como objetivo el diseñar productos interactivos que sean fáciles, efectivos y disfrutables de usar, así como óptimos de la interacción del usuario con un sistema y su entorno. La literatura en DCH ha propuesto abstracciones acerca de los aspectos o principios de diseño como guías de lo que se debe y no se debe hacer al diseñar un sistema [18,19].

Por ello, nuestra investigación evalúa las interfaces propuestas de acuerdo a los siguientes principios básicos de interacción a partir de una matriz ponderada. Este método de evaluación tiene una larga tradición en la ingeniería de software y en la literatura de sistemas de información [20].

Visibilidad. Hacer visibles las interacciones de los humanos con la computadora en la medida de lo posible. Mientras más visibles sean las funciones, más probable será que los humanos realicen la acción apropiada.

Retroalimentación. Proporcionar al usuario información inmediata acerca de la acción que se está ejecutando o que se acaba de ejecutar. En DCH existen diferentes tipos de retroalimentación que involucran el uso de sonidos, animaciones, vibraciones y combinaciones de dichos estímulos sensoriales.

Restricciones. Es limitar las opciones que los usuarios pueden elegir al ejecutar una acción con la finalidad de que el usuario no cometa errores, ofreciendo solo opciones relevantes.

Consistencia. Utilizar operaciones o elementos similares para tareas similares, ya que esto permite que los sistemas sean más fáciles de utilizar porque el usuario aprende un camino aplicable a varios objetos.

Asequibilidad. Indicar o dar pistas acerca de las acciones que se deben de realizar sobre un objeto. Este principio de diseño se ha utilizado mayormente en el diseño de objetos

físicos ya que el mapeo es natural e involucra el uso de metáforas basadas en interacciones reales.

Para la evaluación, se utiliza la siguiente escala de calificación a partir del cumplimiento de los aspectos o principios mencionados anteriormente.

Si cumple: +2 puntos.

Parcialmente cumple: +1 punto.

No cumple: 0 puntos.

Tabla 1. Resultados de evaluación de aspectos de las interfaces del prototipo.

Aspectos	Criterios			
	GUI		NUI	
	Entorno gráfico interactivo 3D	Interfaz de programación grafica por Bloques	Interfaz de interacción kinestésica 3D	Interfaz de interacción por reconocimiento de gestos.
Visibilidad	2	2	2	2
Retroalimentación	2	2	2	1
Restricciones	2	1	1	1
Consistencia	2	2	2	2
Asequibilidad	2	1	2	2

De acuerdo con los resultados de evaluación mostrados en la tabla 1, la **visibilidad** se cumple en las NUI y GUI de la aplicación, el uso de interfaces graficas 3D permiten al usuario observar con claridad el comportamiento de la aplicación con sus acciones. La representación de las manos en 3D y el avatar ayudan a la orientación visual de donde se encuentra el usuario dentro del entorno interactivo, brindan asistencia de cómo actuar en determinadas circunstancias (+2 puntos). La GUI de programación por bloques cumple este aspecto, presenta bloques para la codificación clasificados por tipo en una lista de forma organizada (+2 puntos).

El aspecto de la **retroalimentación** se cumple en la mayoría de las interfaces de la aplicación. En conjunto las NUIs y el entorno interactivo 3D retroalimentan al usuario de sus acciones, a partir de la representación de un modelo 3D de manos en tiempo real (+2 puntos). Por otra parte, la interfaz de reconocimiento de gestos no posee ninguna forma de representación visual, depende de la activación de eventos para producir retroalimentación al usuario (+1 punto). La GUI de programación por bloques, reproduce efectos de audio y retroalimentación visual cuando se conectan los bloques, e indicaciones cuando se está ejecutando código o existe algún error, por lo que cumple con este aspecto (+2 puntos).

Las **restricciones** son limitadas en las interfaces naturales, puesto que su objetivo es el de ofrecer libertad creativa para crear e interactuar, esto ocasiona que no se definan con exactitud las posibilidades de uso de la aplicación (+1 punto). Sin embargo, existen excepciones en la GUI del entorno interactivo 3D, donde su propósito es el de delimitar las dimensiones del escenario y las áreas en las que el usuario puede llegar a interactuar desde la perspectiva de un avatar (+2 puntos).

La **consistencia** se cumple dentro de las GUI, donde el mismo proceso de codificación de escenarios y los eventos es realizado sin importar el tipo de ejercicio, utilizando botones y funciones vistas en otras GUI fáciles de familiarizar como guardar, reproducir, editar y borrar. El usuario puede personalizar como interactúa con el entorno mediante las NUI, pero el proceso de creación e implementación de kinestésia y gestos es natural y siempre es el mismo (+2 puntos).

Las NUI cumplen con el aspecto de la **asequibilidad**, ya que la implantación de reconocimientos de gestos, así como la representación de la mano dentro del entorno interactivo, son por naturaleza, mecanismos de representación de como el humano interactúa con el entorno (+2 puntos). La GUI de programación por bloques, por otra parte, representa metafóricamente a la programación como la unión lógica de distintos elementos relacionados entre sí, los bloques poseen señalamientos de que se pueden unir o ser insertados uno sobre el otro. Sin embargo, no todos los usuarios poseen la capacidad de abstracción para este tipo de conceptos, ocasionando ciertas dificultades de uso (+1 punto).

6. Conclusiones y trabajos futuros

El desarrollo del pensamiento computacional y la búsqueda de mejores formas de educar a las nuevas generaciones sigue siendo una tarea demandante. En este artículo, se presenta un enfoque de las implementaciones de interfaces humano-máquina naturales para el aprendizaje de lógica algorítmica y programación. Aplicando evaluaciones de diseño centrado en el humano, se demuestra que las NUI y GUI presentadas en este proyecto cumplen con los principios básicos de interacción natural, sin embargo, el proceso de DCH es altamente empírico, donde la toma de decisiones es basada en el conocimiento que se tiene de los usuarios y del contexto en el que se utilizará un producto. Como trabajo futuro se planea realizar evaluaciones con grupos de estudiantes de nivel secundaria, para analizar mejor el impacto de la herramienta y medir aspectos de usabilidad, motivación y satisfacción que se experimenta con el uso de este tipo de interfaces.

La aplicación fue desarrollada en JavaScript, utilizando la API de Leap Motion para la recepción de los datos de entrada. La interfaz de programación visual se implementó con Blockly. Para la presentación del entorno gráfico, se utilizó un framework de gráficos 3D para aplicaciones web llamada Babylon JS [21], componente principal para la creación de escenarios tridimensionales, así como la representación de manos humanas dentro del entorno.

Referencias

1. Grover, S., Pea, R.: Computational thinking in K-12: A review of the state of the field. *Educ. Res.* 42(1), 38-43 (2013)
2. Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., Carey, T.: *Human-Computer Interaction*. Essex, UK, UK: Addison-Wesley Longman Ltd. (1994)
3. Lin, W., Du, L., Harris-adamson, C.: *Human-Computer Interaction. User Interface Design, Development and Multimodality*, *Human-Computer Interact. User Interface Des. Dev. Multimodality*, vol. 10271, no. December, pp. 584-592 (2017)
4. Wigdor, D., Wixon, D.: *Brave NUI world: designing natural user interfaces for touch and gesture*. Elsevier (2011)
5. Cabero, J.: Bases pedagógicas del e-learning. *Rev. Univ. y Soc. del Conoc.*, vol. 3, pp. 1-10 (2006)
6. Wenger, E.: Artificial intelligence and tutoring systems: computational and cognitive approaches to the communication of knowledge. *Morgan Kaufmann* (2014)

7. Lin, H.-C. K., Wu, C.-H., Hsueh, Y.-P.: The influence of using affective tutoring system in accounting remedial instruction on learning performance and usability. *Comput. Human Behav.*, vol. 41, pp. 514–522 (2014)
8. Shneiderman, B., Plaisant, C.: *Diseño de interfaces de usuario, Estrategias para una Interacción Pers. Ef. Cuarta Edición* ed Pearson Educ. (2006)
9. Resnick, M. et al.: Scratch: Programming for all. *Commun. Acm.* 52(11), 60–67 (2009)
10. Werner, L., Campe, S., Denner, J.: Children learning computer science concepts via Alice game-programming, p. 427 (2012)
11. Bowe, S.J., Antoniou, M.N., Garrett, C., Huber, B., Kaufman, J., Tarasuik, J.: Young children's transfer of learning from a touchscreen device. *Comput. Human Behav.*, vol. 56, pp. 56–64 (2015)
12. Ebner, M., Spot, N.: Game-Based Learning with the Leap Motion Controller. No. October, pp. 555–565 (2015)
13. Zhu, G., Cai, S., Ma, Y., Liu, E.: A Series of leap motion-based matching games for enhancing the fine motor skills of children with autism. In: *Proc. - IEEE 15th Int. Conf. Adv. Learn. Technol. Adv. Technol. Support. Open Access to Form. Informal Learn. ICALT 2015*, pp. 430–431 (2015)
14. Tran, V.T., Lee, J., Kim, D., Jeong, Y.S.: Easy-to-use virtual brick manipulation techniques using hand gestures. *J. Supercomput.* 72(7), 2752–2766 (2016)
15. Vatavu, R.-D., Anthony, L., Wobbrock, J. O.: Gestures as point clouds, p. 273 (2012)
16. Fraser, N. (et. al.): Blockly: A visual programming editor, URL <https://code.google.com/p/blockly/>, (2013). Último acceso: 2019/05/11
17. Lieberman, H.: *Your wish is my command: Programming by example*. Morgan Kaufmann, (2001)
18. Norman, D.: *The design of everyday things: Revised and expanded edition*. Basic books (2013)
19. Castro, L.A., Rodríguez, M.D.: *A. M. D. E. Computación, Interacción Humano-Computadora y Aplicaciones en México*, Primera ed. Academia Mexicana de Computación, A.C. (2018)
20. Wieringa, R., Maiden, N., Mead, N., Rolland, C.: Requirements engineering paper classification and evaluation criteria: A proposal and a discussion. *Requir. Eng.* 11(1), 102–107 (2006)
21. Catuhe, D., Rousseau, M., Lagarde, P., Rousset, D.: *Babylon.js a 3D engine based on WebGL and javascript* (2014)