

Interfaz de control para un sistema de rehabilitación para niños con síndrome de Down basado en Kinect

Patricia Lorena Irigoyen Sánchez, Jorge de la Calleja, María Auxilio Medina Nieto, Antonio Benitez Ruiz

Universidad Politécnica de Puebla

{patricia.irigoyen, jorge.delacalleja, maria.medina, antonio.benitez}@uppuebla.edu.mx,

Abstract- En este artículo se presenta la implementación de una interfaz de control para un sistema de rehabilitación para niños con síndrome de Down que posean alguna deficiencia en sus capacidades motoras, de lenguaje y de cognición. La aplicación se implementó mediante el Framework XNA Game de Windows el cual cubre funcionalidades para Kinect. Se realizaron tres prototipos de interfaz los cuales fueron evaluados experimentalmente con usuarios potenciales y con los resultados obtenidos se desarrolló la interfaz que atendiera todas las observaciones obtenidas con la evaluación.

palabras clave- síndrome de Down, Kinect, terapias.

I. Introducción

Hoy en día las personas con capacidades diferentes se incorporan a nuestro entorno con mayor facilidad, debido a que se les ofrecen terapias de rehabilitación a temprana edad. Un ejemplo de ello son los niños con síndrome de Down y/o niños que poseen una discapacidad motora.

Para estas personas con alguna discapacidad la meta de llevar una "vida normal" es muy difícil ya que necesitan de atención especial para apoyar sus capacidades de movimiento, conocimiento y lenguaje. Estas terapias no son impartidas por las instituciones de educación básica en México, es por esta razón que se crean fundaciones como FUDAC [10] iniciadas por padres de familia.

Las fundaciones permiten al niño a no permanecer aislado de la educación, así como del sector productivo cuando se convierten en personas adultas. Es entonces que la rehabilitación a temprana edad es necesaria si se desea integrarlos a una vida cotidiana.

El síndrome de Down es una enfermedad en la cual se pueden presentar grados de compromiso motor por lo cual existen distintos grados de terapias para los pacientes [1].

Afortunadamente en la actualidad existen asociaciones/fundaciones que apoyan a estos niños ofreciéndoles ciclos de terapias. Un tipo de terapia es la ocupacional que se enfoca en ayudar a los pacientes a obtener condiciones de vida básicas, como aprender a comer, a vestirse o a sentarse de forma adecuada. En el caso del síndrome de Down, la terapia ocupacional puede emplearse para tratar fuerza muscular débil, para mejorar habilidades motoras y para hacer al niño más independiente.

Por otro lado la tecnología desarrollada en los últimos años permite implementar sistemas que ayuden a terapeutas, médicos y por supuesto a los niños a mejorar sus capacidades de movimiento.

Una de estas tecnologías es el dispositivo Kinect el cual permite crear sistemas como Wake Up¹, que es un sistema de estimulación psicomotriz para niños con síndrome de Down que implementa tecnología ubicua como reloj mental para el desarrollo de la lectura, escritura, habla, capacidad motora, percepción, memoria y atención del paciente. Otro tipo de sistema es el de rehabilitación motriz y cognitiva de pacientes con daño cerebral adquirido² que permite a los pacientes recuperar la funcionalidad perdida o aprender estrategias de compensación.

El uso de la tecnología es de gran ayuda para que los niños puedan integrarse con mayor facilidad a este mundo, particularmente usándola como herramienta de asistencia en sistemas médicos. Es por esta razón que se propone desarrollar un sistema de rehabilitación para niños con síndrome de Down basado en Kinect, con la finalidad de integrarlos a una vida normal.

En el presente artículo se describe sólo el desarrollo de la interfaz de control diseñada para pacientes con síndrome de Down en un rango de 4 a 8 años de edad. La estructura del artículo contempla las siguientes

¹Wake Up es un proyecto de investigación que nace como una contribución de tecnologías ubicuas, como realidad aumentada, visión por computadora y el empleo del Kinect, al desarrollo de los niños con Síndrome de Down. <http://blogs.technet.com/b/microsoftlatam/archive/2012/07/04/conoce-wake-up-el-proyecto-de-per-250-que-va-a-las-finales-de-imagine-cup-2012.aspx>

²Sistema que rehabilitación motora y cognitiva. http://www.salud.es/noticia/Kinect-avance-en-neuror rehabilitacion-del-dano-cerebral#.Ubi_2PlrhN1

secciones: en la segunda se explica de manera general las características del síndrome de Down. En la tercera sección se describe la tecnología Kinect, mientras que en la cuarta se trata sobre el prototipo de la interfaz de control. La quinta sección es dedicada para resultados preliminares obtenidos de las pruebas y por último en la sección seis se presentan conclusiones y trabajo futuro.

II. Síndrome de Down

Fue en el año de 1866 cuando el médico inglés John Langdon Hayden Down, publicó un artículo en el que se describía un síndrome que presentaba características faciales específicas, así como deficiencia en la habilidad motora, deficiencias en el lenguaje y dificultad para la comunicación.

Sin embargo fue hasta el año 1959 en donde se descubrió mediante el análisis de cariotipo³ que estas personas presentaban 47 cromosomas en lugar de 46 y posteriormente se identificó el cromosoma adicional, pequeño y acrocéntrico, correspondiente al par 21. Por lo tanto se estableció que este síndrome, después llamado Síndrome de Down, podía considerarse como una enfermedad, en donde la condición genética es determinada por la alteración del número de cromosomas[3]. Dicho síndrome causa grados de retraso mental y puede presentarse en tres formas principales: la trisomía 21, libre en un 95% de los casos; la translocación, en un 4%; y el mosaicismo, en el 1% de los casos. Estos datos definen el origen y la presencia del síndrome, pero no la cantidad o calidad en que se manifiesta.

Se estima que los factores asociados a esta enfermedad son: consanguinidad, características individuales de los padres, edad materna a partir de los 35 años y edad paterna a partir de los 45 años.

En México las estadísticas del síndrome de Down están a cargo del INEGI que menciona que en el año 2007 una de cada 700 personas presentaba este síndrome, en el año 2008 uno de cada 650, siendo alrededor de 150,000 personas y para el año 2010 el número de casos se agravó siendo ya cerca de 250,000 personas que padecen esta enfermedad. En el estado de Puebla existe una población de 5,778,539 personas de las cuales el 17.44% presenta alguna limitación en la actividad desde el nacimiento captando al síndrome de Down en esta parte de la población.

En el estado de Puebla existen fundaciones como FUDAC (Fundación Down de Puebla AC) la cual tiene su propia escuela registrada ante la SEP, como institución educativa dedicada a la atención de niños

con Síndrome Down con una matrícula de 140 personas de los estados de Puebla y Tlaxcala [12].

En los niños la presencia de este síndrome es alarmante ya que todos tienen algún grado de retraso mental que conlleva a niveles de discapacidad motriz, de lenguaje y/o cognitivo. Dando como resultado una baja probabilidad para que puedan incorporarse a los diferentes sectores como lo es el sector educativo y productivo.

Para que los niños puedan en un futuro ser capaces de desarrollarse en estos sectores es de vital importancia su rehabilitación que requiere una serie de ejercicios o terapias ya sean de tipo motor, cognitivo o de lenguaje, permitiéndoles un avance significativo para su desempeño en la vida cotidiana.

Hoy en día la rehabilitación para estos pacientes se ha ido modernizando ya que emplea tecnología para hacer de la rehabilitación una actividad atractiva y divertida

III. Tecnología Kinect

Kinect es un dispositivo de entrada para la consola Xbox 360 de Microsoft©, siendo una de las consolas de videojuegos más importantes, no sólo para el desarrollo de aplicaciones de entretenimiento, sino también para el desarrollo de aplicaciones en el ramo de las ciencias [5].

Kinect proporciona la posibilidad de ver, oír y procesar instrucciones permitiendo a los usuarios tener una interacción de manera natural con solo hablar y gesticular [5].

Las aplicaciones mediante esta tecnología se han desarrollado también para el apoyo social, especialmente para personas con capacidades diferentes, ya que impulsa al usuario a realizar algún tipo de ejercicio físico que beneficie a su salud.

Las aplicaciones creadas se centran en la recuperación del paciente en aspectos motrices, y cognitivos, de esta manera se ayuda a una integración del paciente con mayor rapidez a una vida cotidiana.

Para desarrollar aplicaciones interactivas Kinect cuenta con componentes como:

- Cámara RGB con una resolución de 640x480 30fps VGA que permite la identificación de cuerpos humanos.
- Cámara y emisor de infrarrojos que permiten realizar un seguimiento del cuerpo en el área de juego.
- Micrófonos que se usan para el reconocimiento de órdenes y charlas
- Motor que permite al Kinect girar y tener un campo de visión de 57 grados en forma horizontal y de 43 grados en forma vertical, además de tener un rango de inclinación física de +/- 27 grados.

³Mapa de los cromosomas en el núcleo de una sola célula. También se le conoce como análisis cromosómico y es usado para diagnosticar algunos tipos de enfermedades genéticas.

IV. Prototipo de la interfaz de control

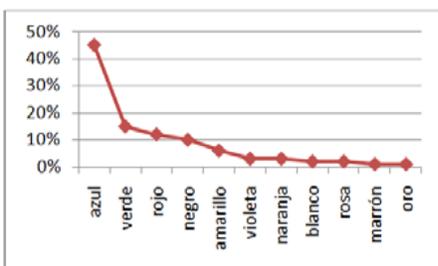
Para desarrollar la interfaz de control se consideraron a las ciencias sociales y a la computación. Las ciencias sociales brindaron información del ambiente, el conocimiento previo, las habilidades, formas y colores que el niño requiere para captar su atención. Por otra parte el campo computacional proporciona métodos para el control de la interfaz. En las siguientes secciones se describen los aspectos considerados para el desarrollo.

A. Aspectos psicológicos para el diseño gráfico de la interfaz de control

Para poder incluir al usuario en el proceso de diseño se aplicó la técnica de “entrevistar y cuestionar”, esto para recabar información acerca de las necesidades y expectativas del paciente; evaluación de alternativas de diseño, prototipos y producto final [2,7].

Los niños con síndrome de Down tienen un cuidado especial debido a las capacidades diferentes que poseen, en términos psicológicos ellos necesitan una estimulación más profunda para realizar cualquier trabajo que un niño haría normalmente [1,3].

Para estimularlos a desarrollar una actividad se debe captar su atención y esto se puede realizar colocando formas, colores, sonidos que sean agradables. De acuerdo a la psicología del color aquellos como el azul, verde, rojo, negro, amarillo, violeta, naranja, blanco, rosa, marrón y oro son los más apreciados. En la Gráfica 1 se muestran porcentajes de agrado de color según [11].



Gráfica 1.- Porcentajes de los colores con más agrado

En cuanto a las formas se detectó que las geométricas van de la mano con la edad de los pacientes; cuadrados, círculos, rectángulos, triángulos son las figuras primarias que los niños a su edad conocen en preescolar.

La manipulación de la interfaz debe ser fácil, es decir, el paciente debe comprender las instrucciones de manera sencilla para ejecutar alguna actividad dentro de la interfaz, por esta razón, las instrucciones se ejecutarán con la mano derecha.

B. Aspectos de funcionalidad para el diseño de la interfaz de control

Para el desarrollo del prototipo de la interfaz de control fue necesario hacer una calibración del dispositivo, es decir, Kinect maneja rangos de accesibilidad [5], dentro de estos rangos el jugador puede usar los controles que se le presentan en pantalla. En la Figura 1 se muestran estos límites.

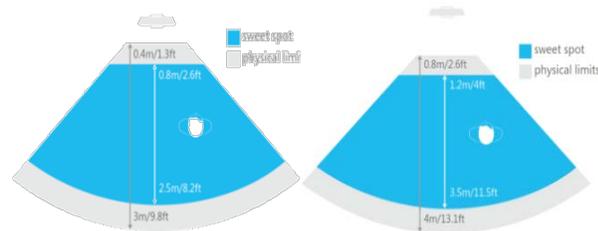


Figura 1.- Rangos de espacio para funcionamiento del Kinect

Para dar inicio al desarrollo de la interfaz de control se creó un nuevo proyecto de tipo Windows Game, al crear un proyecto de este tipo se obtienen los métodos que proporciona XNA Game Studio. XNA es un API para el desarrollo de videojuegos abarcando secciones de Microsoft Game Development incluyendo el kit estándar de desarrollo de Xbox.

En la implementación se inicia con una clase llamada *Game1.cs* la cual tiene la estructura mostrada en la Tabla 1.

```

1. Inicio de clase Game1.cs
   i. Configuración de pantalla
      int anchoV = 1000;
      int altoV = 600;
   ii. Habilitación de sensor Kinect
      Kinect = new Kinect();
   b. Inicializar cada estado de la interfaz de control
      void Initialize(){
         base.Initialize();
         estados.agregar(GameState.principal, new Menu(this));
         estados.agregar(GameState.lenguaje, new lenguaje(this));
         estados.agregar(GameState.Motricidad, new NivelMotriz(this));
         estados.agregar(GameState.Cognicion, new cognicion(this));
         estados.agregar(GameState.sesion1, new Sesiones1(this));
         estados.agregar(GameState.sesion2, new Sesiones2(this));
         estados.agregar(GameState.sesion3, new Sesiones3(this));
         estados.agregar(GameState.extremidades, new Extremidades(this));
         Inicializar el estado inicial estados.iniciar(GameState.principal);
      }
   c. Método para detener el Kinect
      Kinect.parar();
2. Fin clase Game1
    
```

Tabla 1.- Estructura de la clase Game1.cs

En los pasos mostrados en la Tabla 1 se inicializa cada estado de la aplicación, es decir cada uno de los menús con los que se cuenta y ejecuta un método que muestra en pantalla el menú principal de la interfaz de control. Este menú contará con tres botones, cada uno de ellos dará acceso a uno de los módulos, ya sea el módulo de motricidad, de lenguaje o cognición.

Dentro de la implementación existe un método que se encarga de habilitar el dispositivo Kinect mostrado en la Tabla 2.

```

Inicio de clase Kinect.cs
1. instanciar dispositivo Kinect
public KinectSensorDispKinect{
get { return DispositivoKinect; }
set { DispKinect = value; }
}

2. instanciar skeleton
public Skeletonskeleton{
get { returnskeleton; }
set { skeleton = value; }
}

3. obtener movimientos
    a. lectura de coordenadas y proyección de manos

4. inicializar cámara de profundidad kinectSensor.SkeletonStream.Enable(new TransformSmoothParameters{
Smoothing = 0.8f,
Correction = 0.8f,
Prediction = 0.5f,
JitterRadius = 1.5f,
MaxDeviationRadius = 0.9f
});

5. obtener skeleton
skeleton = GetFirstSkeleton(e);

6. iniciar Kinect
Kinect.Start();

7. detener Kinect
Kinect.Stop();

8. fin Kinect
    
```

Tabla 2.- Estructura de la clase kinect.cs

En el método Kinect se encuentra la implementación de la lectura de coordenadas y proyección del movimiento de las manos, para esto es necesario definir el número de frames por segundo, en este caso serán 30.

El método para mover la animación de la mano es el mostrado en la Tabla 3.

```

public void animacionDerecha(Vector2 posManoD) {
textAnimManoD.setSpritePos(new Vector2(posManoD.X - 12, posManoD.Y - 12));
}
    
```

Tabla 3.- Movimiento para animación de la mano

Mientras que el método para proyectar la animación en pantalla es el método mostrado en la Tabla 4.

```

public Vector2 ProyectarD(){
if ((histSkeletonT.Count != 0) && (histSkeletonT[histSkeletonT.Count - 1].Joints[JointType.HandRight].TrackingState == JointTrackingState.Tracked)) {
Joint manoD =
histSkeletonT[histSkeletonT.Count - 1].Joints[JointType.HandRight];
Joint manoDpro =
manoD.ScaleTo(ancho, alto, 0.25f, 0.25f);
}
    
```

```

posmano = new Vector2(manoDpro.Position.X,
manoDpro.Position.Y;
return posmano;
}

else
return new Vector2(10000, 10000);
}
    
```

Tabla 4.- Método para proyección de la mano

Con los métodos presentados en las Tablas 1, 2, 3 y 4 se podrán manipular las secciones siguientes:

- **Menú Motricidad:** El módulo de motricidad presenta dos opciones, una para ejercicios de las extremidades inferiores y la otra para ejercicios de las extremidades superiores. Cada uno de ellos da entrada a las 5 distintas sesiones de movimientos.
- **Menú Lenguaje:** El módulo de lenguaje muestra tres opciones, vocalización, unión de palabras y estructuración de palabras.
- **Menu Cognición:** Este módulo presenta cinco distintos tipos de ejercicios para los pacientes.

Al utilizar el framework XNA se tiene que realizar un *parseo* de cada una de las imágenes que se desee colocar como fondo o como botón. Para hacerlo se utilizó la herramienta XNBFormatter⁴, la cual se muestra en la Figura 3.

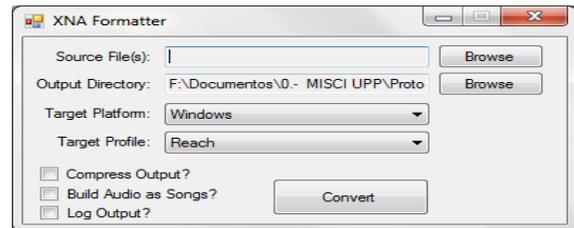


Figura 3.- Herramienta para realizar parseo de una imagen a formato .XNB.

Una vez realizado el *parseo* el archivo *.xnb resultante se traslada a una carpeta contenedora que ya se encuentra definida en el código de la aplicación.

Cuando la aplicación se inicia, de inmediato obtiene los archivos *.xnb definidos en el código de la aplicación para colocarlos en los lugares a los que fueron asignados.

V. Resultados preliminares

Se desarrollaron tres prototipos para la interfaz de control con el objetivo de mostrarlos a un grupo de prueba y evaluarlos. En la Tabla 5 se muestran los resultados obtenidos los tres prototipos.

Da una respuesta teniendo en cuenta que el uno es mala y el 5 es muy bien	Prototipo de Interfaz #1					Prototipo de Interfaz #2					Prototipo de Interfaz #3				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
La aplicación ¿es fácil de manipular?	U1				x					x					x
	U2				x					x					x
	U3				x					x					x

⁴XNBFormatter <http://sourceforge.net/projects/xnbuilder/>

