

Desarrollo e implementación de una interface de usuario interactiva y un módulo de entrenamiento para el sistema de simulación de juego de billar ARPool

Carlos F. Alfaro, Carmen C. Morales
Escuela de Computación
Universidad Don Bosco
carlos.castro@udb.edu.sv
carmen.morales@udb.edu.sv

Resumen

Los avances en la informática y sus aplicaciones en la vida diaria son cada vez más notables. Una de las áreas que se encuentra destacando en este momento es el procesamiento de imágenes, de esta área se desprende la Realidad Aumentada; aplicada en el Sistema ARPool, propiedad intelectual de la Universidad de Queen's Canadá. El presente artículo trata sobre la adición de funciones a dicho sistema, apoyando ese desarrollo en metodología de programación ágil, obteniendo productos más rápidos y acordes a la necesidad del usuario.

Palabras claves: Realidad Aumentada, Visión por computadora, Agile, Software.

Abstract

The advances of informatics and their applications to our daily lives are every day more evident. One of the areas that stands out at present is the application for Augmented Reality, a part of image processing. This application has been used by Queen's University, Canada, to develop the ARPool system. This article describes the addition of new functions to the ARPool system, supporting the development of a versatile programming methodology to obtain products fast and in line with the user's needs.

Keywords: augmented reality, computer's vision, versatile, software.

Introducción

La realidad aumentada, permite definir una visión de la realidad incorporando en ella elementos virtuales, creando de esta forma una realidad mixta en tiempo real. Utilizando Visión por Computadora y Reconocimiento de Objetos, la información sobre el mundo real se convierte en interactiva y digital.

El objetivo del proyecto ARPool (Augmented Reality Pool) es convertirse en un sistema orientado a la mejora de habilidades para los jugadores de billar, proporcionando retroalimentación instantánea.

La propuesta de este trabajo consiste en incorporar dos funcionalidades al sistema existente, logrando con ello la mejora del sistema y una interacción más completa con el usuario. Las funcionalidades a incorporar son: etapa de entrenamiento y menú interactivo.

Para el desarrollo de las mismas se plantea utilizar Desarrollo ágil de software, el cual tiene como principios básico minimizar riesgos desarrollando software en cortos lapsos de tiempo.

Marco Teórico

El primer sistema de Realidad Aumentada fue desarrollado por Ivan Sutherland en 1968, utilizando un dispositivo de despliegue de imágenes tridimensionales tipo casco, para visualizar gráficos tridimensionales (Lara & Villarreal, 2004).

Actualmente, muchos celulares, consolas de videojuegos, PDA y TabletPC ya cuentan con algunos de los dispositivos necesarios para implementar realidad aumentada.

La tradición de interfaces gráficas ha sido la presentación de la mezcla de los mundos virtuales y reales sobre la pantalla y la interacción directa con el mundo real y el virtual.

Existen librerías de software que ayudan al desarrollo de aplicaciones de realidad aumentada, este tipo de aplicaciones implican la superposición de imágenes virtuales en el mundo real.

Antecedentes del Sistema ARPool

El proyecto ARPool (Augmented Reality Pool) es un sistema cámara-proyector que provee retroalimentación en tiempo real a un jugador de billar directamente en la superficie de la mesa (Robotics Computer Vision Lab, 2009).

ARPool es una aplicación que hace uso de visión computacional (una aplicación de visión computacional procesa imágenes adquiridas de una cámara y extrae información de dichas imágenes, tratado de imitar el sistema de visión humano, donde el cerebro procesa las imágenes derivadas de los ojos (Computer visión online, 2012).

Un aspecto importante a considerar es el proceso denominado calibración de cámara, el cual consiste en determinar los parámetros internos de la cámara, tales como distancia focal, factores de distorsión y puntos centrales del plano imagen.

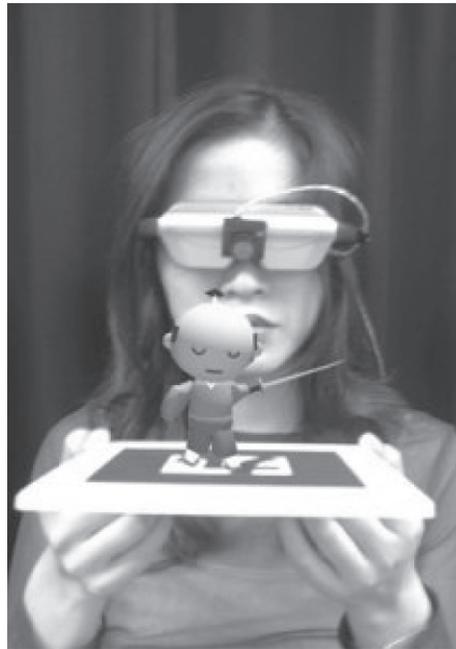


Figura 1 Aplicación utilizando librerías de ARToolkit (ARToolkit, 2012). En la figura aparece un personaje virtual en tres dimensiones de pie en una tarjeta real. Cuando el usuario mueve la tarjeta, el personaje virtual se mueve con él y parece estar asociado.

Sistema de Calibración

En ARPool la calibración es la base para su funcionamiento, pues sin ella no podrá realizar ningún paso posterior; además se debe garantizar que cada vez que se hace una modificación del entorno se recalibre el sistema. Las calibraciones que el sistema requiere son Intrínseca, Extrínseca, homográfica y nuevo modelo de fondo.



Figura 2 Interfaz sistema de calibración ARPool (ARPool user guide, 2010).

Opciones de interfaz:

1. Intrínseca: Describen la geometría y óptica de la cámara.

Elementos a calibrar:

- a. Centro de la imagen
- b. Factores de escala
- c. Todo tipo de aberración
- d. Distancia focal
- e. Enfoque

2. Extrínseca: Definen la orientación y la posición de la cámara respecto a un sistema de coordenadas conocido, al que se suele llamar Sistema de Coordenadas del Mundo.

Elementos a calibrar:

- a. Rotación.
- b. Traslación

3. Homográfica: realiza la calibración del proyector, creando una matriz de transformación llamada homografía que establece la relación entre el plano de la imagen y el plano de la plantilla. Permitiendo dibujar las figuras en la superficie de la mesa de billar en las coordenadas precisas (Viala, 2006).
4. Nuevo modelo de fondo: se realiza una detección de la superficie, indicando que elementos pertenecen al fondo y que no serán tomados en cuenta en el proceso de reconocimiento.

Sistema de Juego

ARPool incorpora detección y rastreo de bola blanca a través de una cámara adherida al techo.

El seguimiento de la señal consiste en extraer los parámetros de tiro, ángulo y desplazamiento de la señal con respecto a la bola blanca, en tiempo real. Estos parámetros son alimentados al simulador del sistema físico el cual calcula las colisiones y las trayectorias de las bolas de billar. (Robotics Computer Vision Lab , 2009)

La interfaz del modo de juego de ARPool (Fig.3) está compuesto por las siguientes áreas:

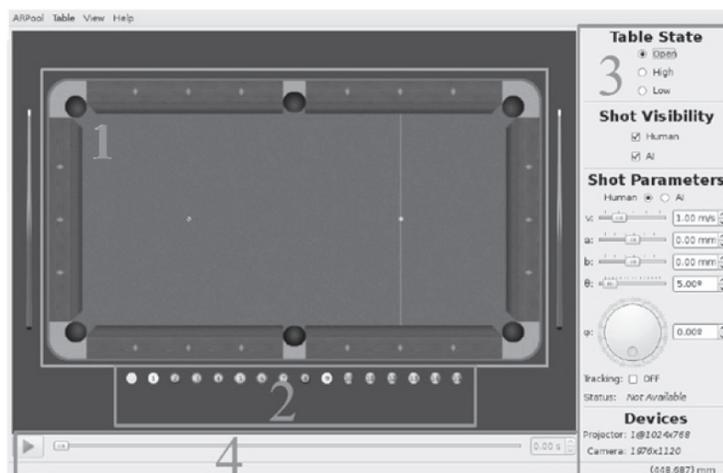


Figura 3 Interfaz de juego ARPool

Esta área es una representación virtual del estado de la mesa de billar. El estado de la mesa se puede configurar para que muestre una representación virtual de la situación actual de la mesa de billar o muestre configure un escenario completamente virtual.

1. En esta área, antes de iniciar el juego aparecen todas las bolas de billar, cuando el juego ha iniciado, solo aparece las que no fueron encontradas en la mesa.
2. En esta área se puede realizar configuraciones de parámetros del juego.
3. Este botón permite realizar una animación del tiro del jugador.

La información inicial proporcionada al jugador de billar, antes de realizar un tiro, incluye:

- Un círculo blanco indicando la posición de la bola blanca sobre la mesa.
- Las bolas de color, estarán dentro de un anillo de luz color verde, indicando su posición sobre la mesa.

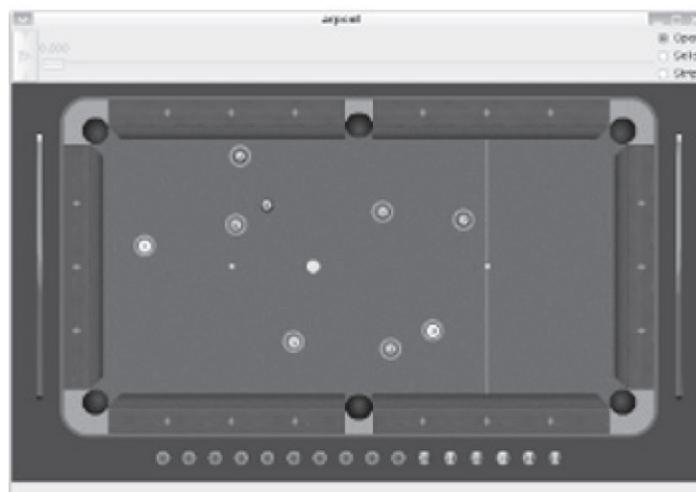


Figura 4 Interfaz virtual de las posiciones de las bolas

- Se mostrarán líneas de color que indicarán las posibles trayectorias de un tiro programado por un algoritmo de Inteligencia Artificial, este algoritmo toma en cuenta las posiciones de las bolas sobre la mesa en cada turno.

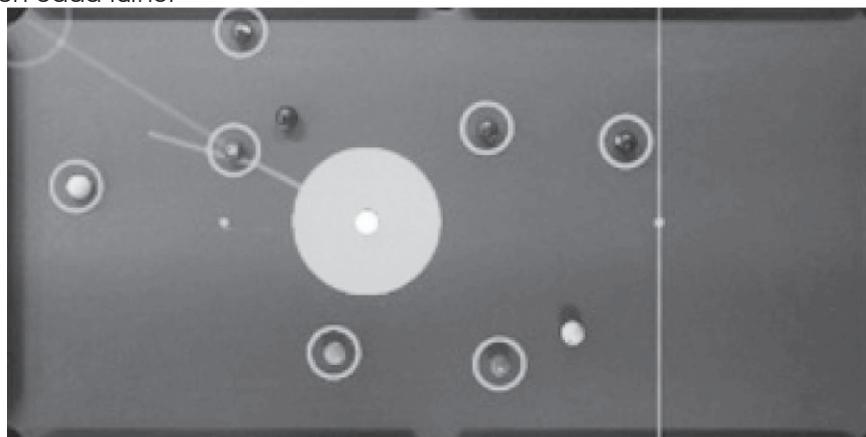


Figura 5 Proyecciones de círculos en ubicación de las bolas sobre la mesa



Figura 6 Interfaz virtual muestra las diferentes opciones de tiro

La retroalimentación proporcionada por el rastro de la bola blanca cuando el jugador está a punto preparando el tiro incluye:

- Líneas de color indicando la ruta de las bolas
- Triángulos blancos indicando los puntos de colisión en los bordes de la mesa
- Círculos de color indicando que bolas estarán involucradas en el tiro.
- Círculos de color en las buchacas indicando si podría entrar o no la bola en la misma

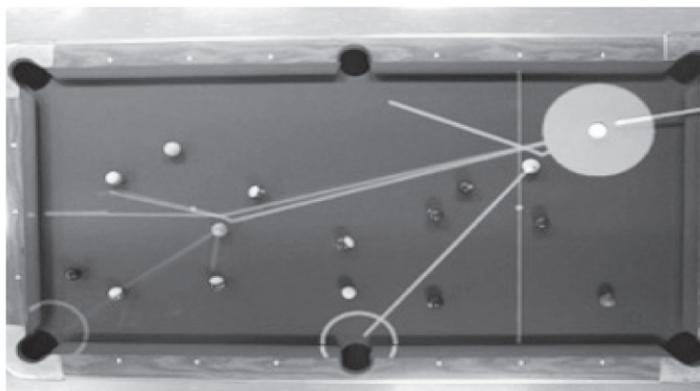


Figura 7 Las proyecciones en la mesa de billar muestra las opciones de tiro

Tecnologías

Visión por Computadora (Computer Vision)

Denominado en español como Visión por Computadora, Visión Artificial o Visión Técnica (Unzueta, 2009), es la transformación en datos provenientes de una cámara fija o de video, estos datos son utilizados para toma de decisiones o para nuevas representaciones (Garay & Kaehler, 2008) Los datos obtenidos pueden o no contener información del entorno, personas involucradas, sitio de montaje, entre otros.

En los sistemas de visión de computadora, se reciben una red de números provenientes de la cámara o del disco. Lo que se busca lograr es recrear la visión en 3D (tres dimensiones) que el ojo humano es capaz de distinguir. (Szeliski, 2011) Para ello los investigadores del tema han desarrollado de forma paralela técnicas matemáticas para su realización.

Hoy en día la Visión por Computadora tiene múltiples aplicaciones: Construcción de modelos 3D, Imágenes Médicas, Seguridad de automóviles, Inspección de maquinaria, reconocimiento de huellas digitales y biométricas, entre otros.

OpenCV

OpenCV fue desarrollado originalmente por Intel; pero actualmente es una librería de visión por computadora de código abierto, la cual ha sido escrita en los lenguajes de programación C y C++, trabajando bajo las plataformas Linux, Windows y Mac OSX. (Parker , 2011)

El fuerte principal de OpenCV es su diseño orientado a buscar la eficiencia en el uso de la computadora y su uso en aplicaciones en tiempo real. Debido a que fue desarrollada en un optimizado C, puede sacar mayor provecho de procesadores de múltiple núcleo.

Uno de los propósitos de su construcción es proporcionar una infraestructura de visión de computadora de uso simple que ayude a los investigadores y desarrolladores a crear sofisticadas aplicaciones de visión de una forma rápida. (Agam, 2012)

OpenGL

OpenGL es una interfaz de software para el hardware de gráficos. Esta completa interfaz consiste en alrededor de 120 distintos comandos, los cuales se utilizan para especificar los objetos y las operaciones necesitadas para crear aplicaciones interactivas en tres dimensiones.

OpenGL está diseñado para funcionar de forma eficiente, incluso si la computadora utilizada no tiene el hardware adecuado para desplegar el programa de gráficos. (Shreiner, 2004)

Su diseño lo permite ser ágil, independiente del hardware de interfaz y precisamente esto permite que sea implementado en muchas plataformas de hardware diferentes; sin embargo no proporciona comandos de alto nivel para la descripción de modelos de objetos tridimensionales. Para ello deberá construirse el modelo deseado con un pequeño conjunto de primitivas geométricas. Por lo cual incluye una biblioteca sofisticada que incluye estas características.

Realidad Aumentada (Augmented Reality)

La realidad aumentada intenta integrar elementos virtuales dentro de un entorno real (Bimber & Raskar, 2005) Añade o escribe elementos dentro de la misma, esta es su principal diferencia con la realidad virtual, ya que no sustituye sino que añade.

El reto dentro de esta área es que el entorno real es más difícil de controlar que uno completamente virtual. La información que desee añadirse al entorno debe tener un fuerte vínculo con la misma, este se basa principalmente en la relación espacial entre lo virtual y lo real.

Las características que distinguen a la realidad aumentada son (Madden, 2011)

- a) Combinación del mundo real con gráficos de computadora
- b) Proporcionar interacción con objetos en tiempo real
- c) Pistas de objetos en tiempo real
- d) Proporcionar reconocimiento de imágenes u objetos
- e) Proporciona datos o entornos en tiempo real

Los elementos básicos que permiten una realidad aumentada son los siguientes (Rolf , 2009):

- a) Dispositivo de vídeo: Necesario para capturar información del mundo real, con el cual se alimenta el sistema.
- b) Unidad de procesamiento: Esta dependerá del requerimiento del sistema a desarrollar, puede variar desde un teléfono inteligente hasta una computadora de mesa. Sin embargo lo importante a tomar en cuenta será el cálculo de la CPU (registro de la realidad, lógica de la aplicación) y la GPU (renderizado de los objetos virtuales).
- c) Dispositivo de visualización: la imagen en donde se combina el mundo real con el virtual debe mostrarse en un dispositivo de visualización.

Tecnologías complementarias

Adicional a las tecnologías principales expuestas, existen otras herramientas que se utilizan y que han sido de apoyo o soporte para la utilización de las principales.

1. C++

Lenguaje de programación diseñado en 1980. Su creación se dio con el objetivo de extender el lenguaje de programación C incluyendo manipulación de objetos. (Stroustrup, 2001)

2. QT SDK

Es una herramienta multiplataforma ampliamente usada para desarrollar aplicaciones con una interfaz gráfica de usuario. (Summerfield , 2010)

3. Bazaar

Sistema de control de versiones distribuido, diseñado para facilitar la contribución de proyectos de software libre y opensource. Puede ser usado por un usuario trabajando múltiples ramas de un contenido local, colocado en red puede ser manipulado por un grupo de trabajo. (Raymond., 2011)

4. Linux Ubuntu

Sistema operativo, desarrollado por comunidades de programadores. Utiliza núcleo Linux y su origen proviene de Debian. Ubuntu ha sido desarrollado con la visión de ser aceptado por el usuario promedio (Sobell, 2010).

Metodologías de Desarrollo Ágil

Metodologías de Desarrollo Ágil de Software

El objetivo de estas metodologías es permitir a los equipos de desarrollo, crear rápidamente y responder de forma oportuna a los cambios que surgen en los proyectos. (Abrahamsson & Salo & Ronkainen & Warsta, 2002)

The Agile Alliance, es una organización dedicada a promover los conceptos relacionados con desarrollo de software. De acá surge el manifiesto Ágil, documento que resume la filosofía.

Según este documento lo que debe valorarse en un proyecto son:

- a) Los individuos y la interacción por encima de los procesos y herramientas.
- b) El software que funciona por encima de la documentación excesiva.
- c) La colaboración con el cliente por encima de la negociación contractual.
- d) La respuesta al cambio por encima del seguimiento de un plan.

Existen diversas metodologías de desarrollo ágil de software; sin embargo todas mantienen constantes en común como: son metodologías simples y fáciles de aprender, los equipos de trabajo no son jerárquicos y están auto-organizados, comunicación intensiva, minimalismo, proceso iterativo e incremental, capacidad adaptativa.

A continuación se explican brevemente las tres metodologías principalmente difundidas y sobre todo utilizadas por equipos desarrolladores de software.

1. Programación Extrema (Extreme Programming –XP)

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software (Wesley, 1999). XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

La programación extrema se encuentra organizada en tres apartados:

- a. La historia del usuario: Son las técnicas para especificar los requerimientos del software. En la práctica son tarjetas de papel en las cuales el cliente describe brevemente características que el sistema debe poseer (el equipo decidirá si son funcionales o no). Cada historia debe ser delimitada y comprensible para ser realizada en poco tiempo.
- b. Roles: En la propuesta original realizada se contemplan siete roles; programador, cliente, encargado de pruebas, encargado de seguimiento, entrenador (coach), consultor, gestor.
- c. Proceso: El ciclo del desarrollo ideal con XP consisten en seis fases; Exploración, Planificación de la Entrega, Iteraciones, Producción, Mantenimiento y Muerte del Proyecto.

2. Scrum

Es óptima para proyectos con rápido cambio de requisitos (Schwaber & Beedle & Martin, 2001). Enfatiza un conjunto de valores y prácticas administrativas, en lugar de darle relevancia a requisitos o implementación. Sus dos principales aspectos:

- a. El desarrollo de software se realiza mediante iteraciones, llamadas sprints, con una duración de 30 días. El resultado de cada iteración es un incremento ejecutable que se muestra al cliente.

- b. Se programan reuniones diarias durante la vida del proyecto (con un promedio de quince minutos por reunión), en estas reuniones se busca coordinación e integración del equipo.

Los roles contemplados en esta metodología son: Scrum master, propietario del equipo, equipo scrum, cliente, administración.

3. Dynamic Systems Development Method (DSDM)

Define el marco para desarrollar un proceso de producción de software (Stapleton, 1997). Nace en 1994 con el objetivo de crear una metodología RAD unificada. Sus principales características son: es un proceso iterativo e incremental y el equipo de desarrollo y el usuario trabajan juntos.

Propone cinco fases: estudio viabilidad, estudio del negocio, modelado funcional, diseño y construcción, y finalmente implementación. Las tres últimas son iterativas, además de existir realimentación a todas las fases.

Selección de la metodología a utilizar

Aunque no existe una metodología que se recomiende para todos los proyectos, puede seleccionarse una de ellas de acuerdo a la naturaleza, envergadura, recursos disponibles y requerimientos del proyecto. Extreme Programming puede ser muy útil y efectivo para trabajar con C++ (Jeffries, 2004)

Adicional a esto, se han seleccionado criterios de acuerdo al proyecto que se está desarrollando:

- a. Cantidad de información disponible sobre la metodología y es el más difundido (Gómez & Rosale & Salas, 1993)
- a. Extreme Programming goza de mucha popularidad en cuanto a uso y documentación. Esto se debe principalmente a su facilidad de implementación y manejo de iteraciones en cortos períodos de tiempo.
- b. Conocimiento, aplicación y uso previo del equipo. (Kniberg, 2007)
- c. El equipo de trabajo encargado de realizar la adición de las funcionalidades del sistema ARPool tiene mayor información sobre cómo se trabaja con Extreme Programming, adicional a ello han tenido experiencias de trabajo con esta metodología. Además otro de sus fuertes es que está orientado a propiciar el buen trabajo en equipo.
- d. Adaptado a 40 horas de trabajo semanales.
- e. Uno de los principios fundamentales de la metodología es que sugiere cuarenta horas de trabajo semanal. Además de no sobrecargar al equipo de trabajo con actividades que no puedan realizarse en los plazos estipulados o incluso deriven en la desmejora de la calidad de las entregas.
- f. Procesos iterativos cortos y entregas en períodos establecidos
- g. Los principios de esta metodología mantienen que los procesos iterativos no deben exceder de 3 semanas, dependiendo de la magnitud y complejidad de la entrega. Los períodos son respetados pues permiten la retroalimentación del cliente (supervisor) permitiendo realizar mejoras o dando el aval para un nuevo versionamiento.
- h. Enfocado a grupos pequeños
- i. El grupo de trabajo requerido es bastante reducido, e incluso una misma persona podría asumir dos roles en determinado momento. Dentro de la programación también promueve el desarrollo en parejas, pues esto permite la alternancia e incluso dos puntos de vista diferentes sobre cómo tratar un problema; dando como resultado un software de mayor calidad.

- j. El grado de involucramiento del cliente (en este caso el supervisor del proyecto)
- k. Uno de los roles utilizados y una de las filosofías que promueve XP, es un alto grado de participación del cliente, pues debe estar presente para la definición de requerimientos, entrega de avances y sus opiniones proporcionan al equipo de programación ideas sobre cómo mejorar las funcionalidades.

Por todo lo anterior, la metodología seleccionada es: Extreme Programming (XP).

Diseño e Implementación de Menú interactivo y fase de entrenamiento Basado en la Metodología Elegida

Las dos mejoras que se desarrollarán permitirán una mejor interacción con el sistema, llevar el proyecto a un nivel de entrenamiento que al mismo tiempo resulte divertido para principiantes en el juego.

1. Menú interactivo.

El menú interactivo busca eliminar la dependencia de la computadora para ejecutar las funciones propias al juego. Etapas como Correr la aplicación (RUN), Entrenamiento (TRAIN) y pedir ayuda al sistema (AI QUERY) serán manejadas directamente desde la mesa de billar, de esta forma se agiliza la ejecución del sistema y resulta más atractivo para los usuarios. El menú interactivo se muestra se muestra en la figura 8.

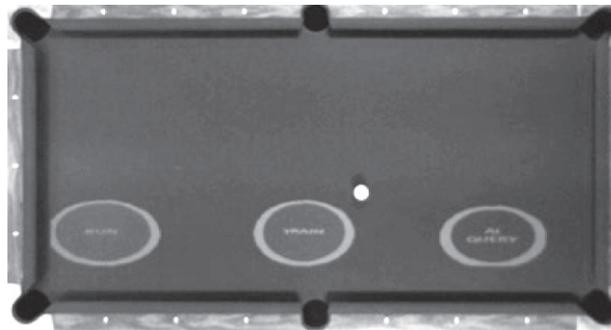


Figura 8 Menú interactivo desplegado en mesa de billar, las opciones desplegadas son: RUN, TRAIN y AI QUERY

Se planea almacenar imágenes previas de las opciones del menú, estas servirán como máscaras que serán comparadas con la foto posterior y con ello se logrará detectar la diferencia y donde se encuentra ubicada la mano.

El sistema cámara, proyector, espejos y computadora permitirá que el menú diseñado sea desplegado en la mesa y al aparecer las opciones sobre la misma, el usuario determine con su mano, cuál de ellas seleccionará. Después de ello el sistema deberá tomar una fotografía y comparar las máscaras versus la nueva imagen. Ubicando así donde está la diferencia y por ende la mano.

2. Fase de Entrenamiento

Dicha fase busca mejorar la habilidad en el juego para personas no expertas en la materia, las facetas que se propone entrenar son las de potencia e intensidad de tiro, dirección, precisión.

Esta fase consistirá en una serie de Ejercicios dónde se hará uso de la bola blanca y una serie de círculos que se dibujarán a medida se desarrolle. Entre la bola y el círculo aparecerá una línea que indicará al usuario que dirección deberá seguir para que la bola llegue hasta donde está el círculo.

El entrenamiento contará con una puntuación, que se calculará a partir, de la distancia entre la bola y el círculo y el número de intentos del usuario. Esto se muestra en las figures 10, 11 y 12

Para llevar a cabo lo anterior se utilizará como apoyo el desarrollo ágil de software, específicamente Extreme Programming.

XP es una metodología basada en abordar las limitaciones en el desarrollo de software. No se ocupa de la gestión de la cartera de proyectos, la justificación financiera de los proyectos, operaciones, marketing o ventas. XP tiene implicaciones en todas estas aéreas, pero no aborda directamente estas prácticas.

Esta metodología se utiliza para minimizar riesgos en el desarrollo de software en cortos lapsos de tiempo y de igual forma permite a los equipos de desarrollo responder de forma oportuna a los cambios que surgen en los proyectos.

Las etapas del proyecto se abordan según las fases de la metodología ágil:

Fase I: Exploración.

En esa etapa se levantan las historias de usuario así como también al mismo tiempo los desarrolladores se familiarizan con las tecnologías, herramientas y prácticas utilizadas en el transcurso del desarrollo del proyecto.

- Historias de Usuario¹.

Para registrar los requerimientos del cliente se utilizaron las fichas sugeridas por la metodología XP para recopilar las historias de usuario y clasificar las tareas que resultantes de las historias.

Historia 1: Modo de entrenamiento

Tarea 1.1: Diseñar diagrama de secuencia para modo de entrenamiento

Tarea 1.2: Reconocimiento inicial de la mesa de billar

Tarea 1.3: Determinar distancia entre círculo y bola blanca

Historia 2: Menú interactivo

Tarea 1.1: Diseñar diagrama de secuencia para menú interactivo

Tarea 1.2: Creación de menú y proyección del mismo en la mesa de billar.

Tarea 1.3 Determinación de opción seleccionada

Tarea 1.4: Proyección de función seleccionada y modo de espera de menú.

Fase II: Planificación

En esta fase se establece la prioridad de cada historia de usuario y se realiza una estimación de esfuerzo para cada una de ellas. Las estimaciones de esfuerzo están asociadas a la implementación de las historias, utilizando como medida el punto, entendiéndose como punto a una semana ideal de programación.

¹ El detalle las historias y tareas se encuentra en las tablas 1-9 en el apéndice 1.

B.1 Tabla de historia de usuarios y puntos.

Iteración	No	Historias	Inicio	Fin	Observaciones
Primera	1	Modo de entrenamiento	09/12/2012	10/12/2012	
Segunda	2	Menú interactivo	10/01/2012	11/012012	

B.2 Cronograma de desarrollo por iteraciones

No	NOMBRE	Diciembre 2011				Enero 2012			
		S1	S2	S3	S4	S1	S2	S3	S4
1	Exploración								
2	Iteración 1								
3	Iteración 2								

Fase III: Iteraciones.

Esta fase incluye las iteraciones sobre el sistema antes de ser entregado. Todo el trabajo de la iteración es expresado en tareas, cada una de ellas es asignada a un programador como responsable, pero llevadas a cabo por parejas de programadores.

C.1 Iteración 1

Historia de usuario 1: Modo de entrenamiento

Tarea 1. Diseñar diagrama de secuencia para modo entrenamiento

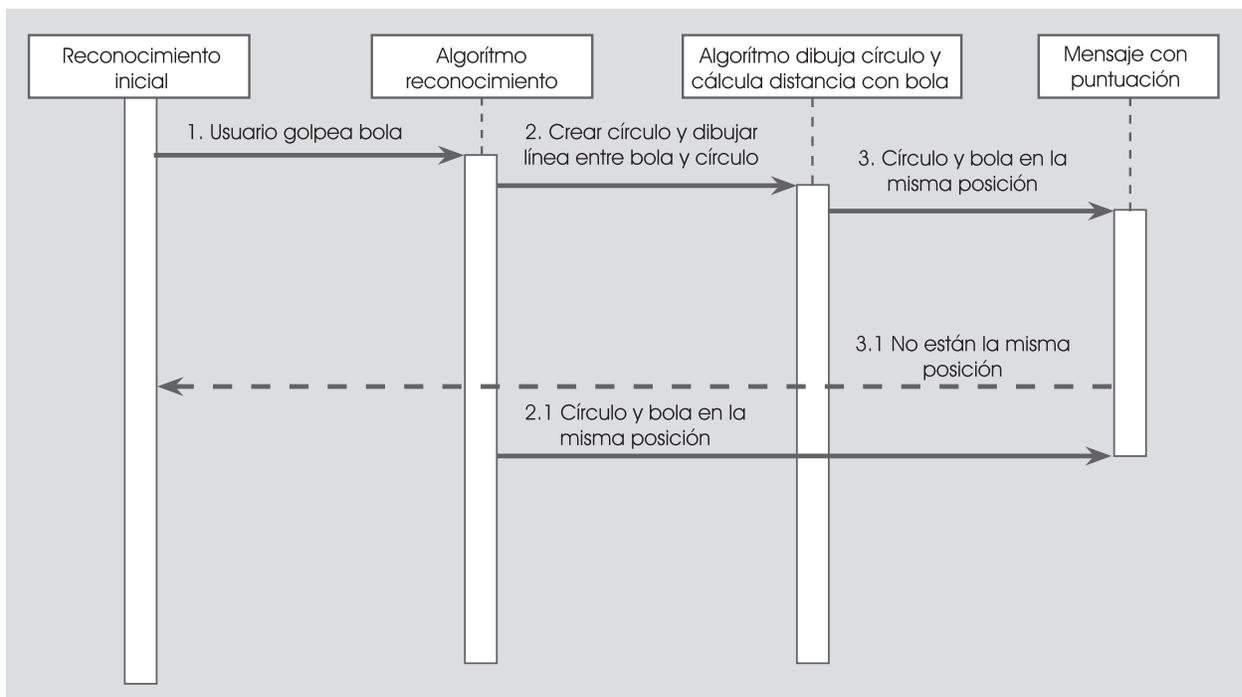


Figura 9 Diagrama de secuencia modo entrenamiento

Procesos de Modo entrenamiento:

1. El sistema mostrará la ubicación inicial de la bola y el círculo, el usuario deberá golpear la bola.
2. El sistema ejecutará un algoritmo de reconocimiento para detectar la bola blanca.
3. Luego de haber encontrado la bola blanca, procederá a dibujar un círculo nuevamente sobre la mesa y procederá a calcular la distancia entre el círculo y la bola blanca.
4. Si el usuario entra al área del círculo, la distancia entre ellos será de 0 y el sistema enviará un mensaje con una puntuación, que será calculada mediante el número de oportunidades que el usuario utilizó para poder ingresar la bola blanca en el círculo.

Tarea 2: Reconocimiento inicial de la mesa de billar.

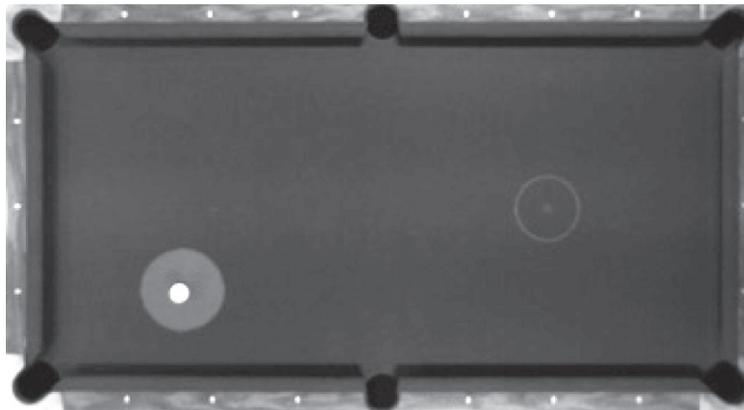


Figura 10 Reconocimiento inicial y proyecciones en mesa

Tarea 3: Determinar distancia entre círculo y bola blanca

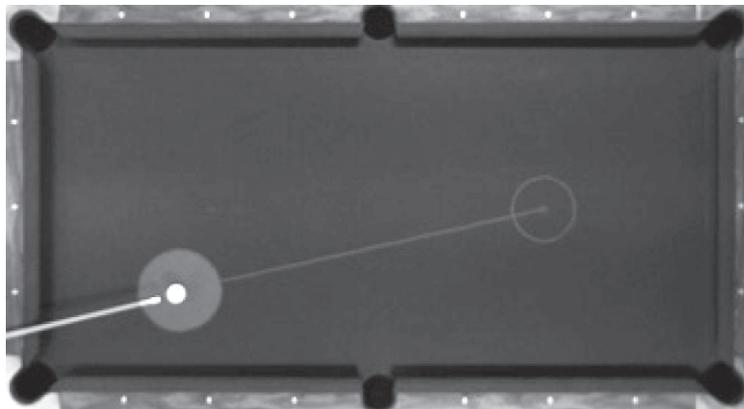


Figura 11 Muestra de trayectoria y cálculo de distancia entre círculo y bola blanca

Si la bola blanca entra en el círculo, se detiene el ejercicio y muestra la puntuación. Como se muestra en la figura 12.



Figura 12. Etapa final. Si la bola está dentro del círculo, se muestra un mensaje con la puntuación.

C.2 Iteración 2

Historia de usuario 2: Menú Interactivo.

Tarea 1. Diseñar diagrama de secuencia para menú entrenamiento.

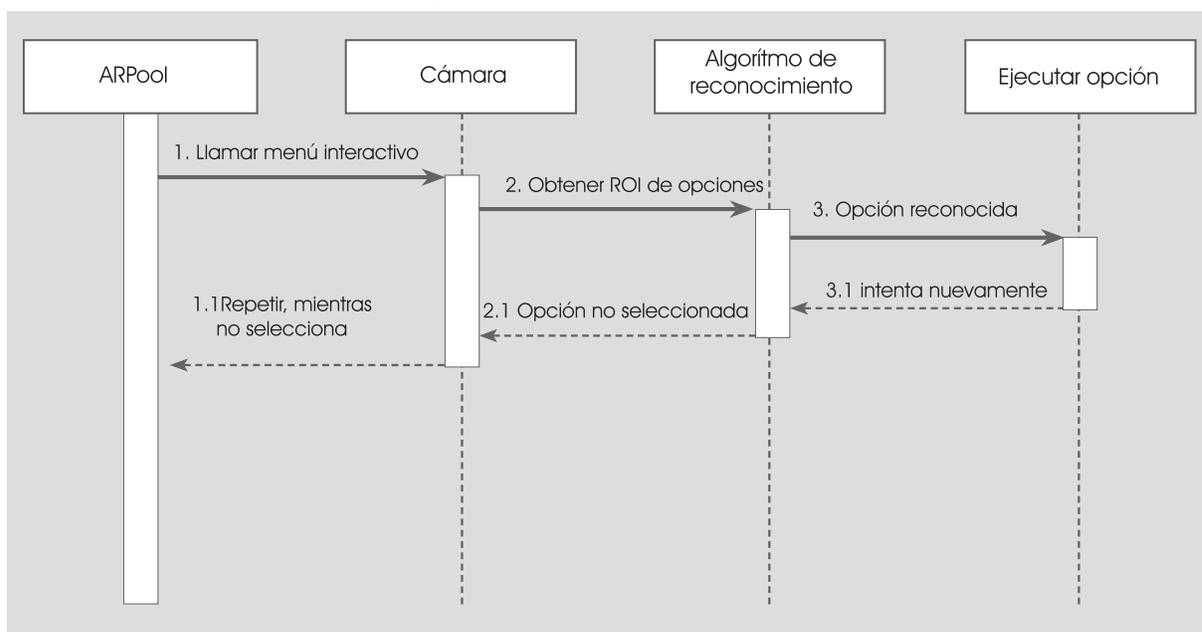


Figura 13 Diagrama de secuencia Menú interactivo

Procesos de menú interactivo:

1. Al iniciar el sistema, se mostrará el menú interactivo.
2. Se estarán realizando capturas de la mesa de billar, esperando que exista un cambio en ella.
3. Se extraerán las opciones del menú, de la imagen adquirida con la cámara y se procederá a reconocer, la opción seleccionada.
4. En el momento de reconocer una opción se procederá con la ejecución de la misma.

Tarea 2: Creación de menú y proyección del mismo en la mesa de billar

El menú de la tarea 2 se encuentra reflejado en la figura 8.

Tarea 3: Determinación de opción seleccionada.



Figura 14 Usuario selecciona una opción y sistema muestra mensaje con opción seleccionada y la ejecuta.

Fase IV: Pruebas de aceptación del cliente

Estas pruebas permiten que el cliente se asegure que se ha desarrollado la funcionalidad solicitada durante la fase de exploración. Las pruebas se realizan en base a parámetros, denotados por letras y dentro de cada parámetro se encuentran las pruebas realizadas denotadas por número.

D.1 Pruebas de aceptación para la historia de usuario 1. Fase de entrenamiento.

- i. Identificar todos los posibles resultados observables de la historia*
 - o Reconocimiento de estado inicial de la mesa
- ii. Identificar los resultados que terminan la historia y los que permiten continuar dentro de la historia*
 - o Historia concluye cuando la bola blanca termina dentro del círculo
 - o La historia continua mientras la bola blanca permanezca fuera del círculo
- iii. Identificar la ejecución de caminos posibles*
 - o Se reconoce la bola blanca y se calcula la distancia con respecto al círculo.
 - o Se válida si la distancia entre el círculo y la bola es cero.
 - o Cuando la distancia es igual a cero envía un mensaje con la puntuación obtenida
 - o Cuando la distancia no es cero inicia nuevamente el proceso de reconocimiento y aumenta el contador de número de tiros.
- iv. Asignar un conjunto de valores válidos y valores del entorno a cada miembro de ejecución para obtener el resultado esperado*
 - o Los valores se obtienen a partir del reconocimiento de la ubicación de la bola blanca
- v. Eliminación de caminos redundantes*
 - o No existen caminos redundantes

D.2 Pruebas de aceptación para la historia de usuario 2. Menú interactivo.

- i. Identificar todos los posibles resultados observables de la historia*
 - o Menú proyectado sobre la superficie de la mesa
- ii. Identificar los resultados que terminan la historia y los que permiten continuar dentro de la historia*

- o Reconocimiento de una de las opciones
- o Continua mientras el sistema no reconozca una opción válida
- iii. Identificar la ejecución de caminos posibles
 - o Ejecución de función RUN
 - o Ejecución de función TRAIN
 - o Ejecución de función AI Query
- iv. Asignar un conjunto de valores válidos y valores del entorno a cada miembro de ejecución para obtener el resultado esperado.
 - o Sólo se aceptará una opción a la vez, en caso de intentar ingresar dos valores el sistema se mantendrá en estado de espera.
- v. Eliminación de caminos redundantes
 - o No existen caminos redundantes

Conclusiones

- Para desarrollo del menú interactivo se hizo uso de las librerías openCV, específicamente las funciones que están relacionadas con sustracción de imágenes y la coincidencia de plantillas.
- Se logró determinar que XP es la metodología ágil que más se adecua a las necesidades de este proyecto.
- La creación del menú logró mejorar la interacción, comodidad y facilidad de uso de sistema porque se independizó del uso la computadora, utilizando únicamente el menú interactivo.
- El modo de entrenamiento permitió que el sistema ARPool, sea más entretenido para los usuarios de diferentes niveles de experiencia en el juego.
- La realidad aumentada permite una percepción más completa de la realidad, proporcionando información adicional a través de proyecciones holográficas sobre objetos reales.

Referencias

- Abrahamsson, P & Salo, O. & Ronkainen, J & Warsta, J. (2002) *Agile software development methods Review and analysis..* VTT Publications.
- Agam G. (2012) *Introduction to Programming with OpenCV*. Autor: Department of Computer Science. Illinois Institute of Technology.
- ARToolkit(2012) Sitio web del proyecto ARToolkit. Disponible en línea: <http://www.hitl.washington.edu/artoolkit/>.
- Bimber O. & Raskar R. (2005) *Spatial Augmented Reality. Merging Real and Virtual Worlds*.
- Computer visión online (2012). Sitio web del proyecto. Disponible en línea: <http://www.computervisiononline.com/books/computer-vision/introduction>. Verificado 21 de enero 2012
- Garay B. & Kaehler A. (2008) *Learning OpenCV*. Editorial O'Reilly Media, Inc. Estados Unidos. Primera Edición.
- Gómez T & Rosales O & Salas J (1993). *Criterios de selección de metodologías de desarrollo de software*. Ind. data, jul. 2010, vol.13, no.2, p.70-74. ISSN 1810-9993.
- Jeffries R. (2004) *Extreme Programming adventures in C#*. MS Press,
- Kniberg H. (2007) *Scrum y XP desde las trincheras*. InfoQ Enterprise Software Development Series.
- Lara L. & Villarreal J. (2004) "La Realidad Aumentada: Una Tecnología en espera de usuarios." *Revista Digital Universitaria de Universidad Autónoma de México* Volumen 5 número 7.

- Madden L. (2011) *Professional Augmented Reality Browsers for SmartPhones. Programming for Junaio, Layar and Wikitude*. Editorial Wrox. United Kingdom. Primera Edición.
- Parker J. (2011) *Algorithms for Image Processing and Computer Vision*. Editorial Wiley Publishing Inc. Segunda Edición.
- Queen's University (2010). *ARPool user guide*. ECE Department.
- Raymond. E. (2011). *The Cathedral & The Bazaar. Musings on Linux and Open Source by an Accidental Revolutionary*.
- Robotics Computer Vision Lab (2009). Sitio informativo de proyecto ARPool. Disponible en línea: <http://rcvlab.ece.queensu.ca/~qridb/ARPOOL.html>. Verificado 21 de enero 2012
- Rolf R. (2009) *The end of hardware. Augmented Reality and Beyond*. Tercera Edición. Editorial BookSurge.
- Schwaber K., Beedle M., Martin R.C. (2001) *Agile Software Development with SCRUM*. Prentice Hall.
- Shreiner D. (2004) *Open GL Reference Manual. The oficial Reference Document to OpenGL, versión 1.4*. Cuarta Edición.
- Sobell M. (2010) *A practical Guide to Linux Ubuntu* Tercera Edición.
- Stapleton J. (1997) *Dsdm Dynamic Systems Development Method: The Method in Practice*. Addison-Wesley. 1997.
- Stroustrup B. (2001) *The C++ Programming Language*. Pearson Education. Tercera edición.
- Summerfield M. (2010) *Advanced Qt Programming. Creating great software with C++ and Qt 4*. Prentice Hall Open Source Software Development Series.
- Szeliski R. (2011) *Computer Vision: Algorithms and Applications*. Editorial Springer-Verlag. British.
- Unzueta L. (2009) Tesis doctoral "Captura sin marcadores del movimiento del cuerpo humano y reconocimiento de acciones combinadas para la interacción hombre computador". Universidad de Navarra, España. <http://www.tecnun.es/investigacion/tesis/t09/meca72.html>. Disponible en línea.
- Viala C. (2006) "Caracterización y optimización del proceso de calibrado de cámaras basado en plantilla bidimensional" Tesis doctoral, Universidad Politécnica de Valencia. Capítulo 2: El modelo de la cámara-2.7 Homografías pag. 29.
- Wesley A. (1999) Beck, K. "Extreme Programming Explained. Embrace Change", Pearson Education, Traducido al español como: "Una explicación de la programación extrema. Aceptar el cambio"

Apéndice 1

Historia de Usuario	
Número: 1	Usuario: supervisor
Nombre Historia: Modo de entrenamiento	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 3	Iteración asignada: 1
Descripción: Tiene como fin, mejorar las habilidades del usuario en el juego de billar. Haciendo uso de ejercicios prácticos, en los que el usuario deberá medir su fuerza y calcular los tiros que solicite el sistema	
Observaciones	

Tabla 1. Historia de de usuario 1

Tarea	
Número: 1.1	Número historia: 1
Nombre tarea: Diseñar diagrama de secuencia para modo de entrenamiento	
Tipo de tarea: Diseño	Puntos Estimados: 1/8
Fecha inicio:10/12/2011	Fecha fin:11/12/2011
Responsable: Carlos Alfaro	
Descripción: se realiza el diseño de diagrama de secuencia	

Tabla 2. Tarea 1.1 Historia de Usuario 1

Tarea	
Número: 1.2	Número historia: 1
Nombre tarea: Reconocimiento inicial de la mesa de billar	
Tipo de tarea: Desarrollo	Puntos Estimados: 2
Fecha inicio:12/12/2011	Fecha fin:25/12/2011
Programador responsable: Carlos Alfaro	
Descripción: Se modifica la función de reconocimiento de bolas de billar, para que únicamente detecte la bola blanca; posteriormente dibujará un círculo a su alrededor para indicar su posición. Dibujar un círculo en la mesa donde se deberá ingresar la bola blanca.	

Tabla 3. Tarea 1.2 Historia de Usuario 1

Tarea	
Número: 1.3	Número historia: 1
Nombre tarea: Determinar distancia entre círculo y bola blanca	
Tipo de tarea: Desarrollo	Puntos Estimados: 1
Fecha inicio:26-12-2011	Fecha fin:30-12-2011
Programador responsable: Carlos Alfaro	
Descripción: Se desarrolla un algoritmo que calcule la distancia entre la bola blanca y el círculo dibujado, el algoritmo se mantendrá calculando esta distancia hasta que la bola blanca esté dentro del círculo. Se obtendrá una marcación, según el número de oportunidades que utilizó para lograr su objetivo.	

Tabla 4. Tarea 1.3 Historia de Usuario 1

Historia de Usuario	
Número: 2	Usuario: supervisor
Nombre Historia: Menú interactivo	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados:3	Iteración asignada:2
<p>Descripción:</p> <p>Facilitar la interacción del usuario con el sistema, mediante la proyección de las opciones más utilizadas de ARPool sobre la mesa de billar. El sistema menú deberá esperar a que el usuario seleccione una opción, posteriormente el sistema deberá ejecutar la opción indicada por el usuario</p>	
Observaciones:	

Tarea 5. Historia de Usuario 2

Tarea	
Número: 2.1	Número historia: 2
Nombre tarea: Diseñar diagrama de secuencia para menú interactivo	
Tipo de tarea: Diseño	Puntos Estimados: 1/8
Fecha inicio:02/01/2012	Fecha fin:03/01/2012
Responsable: Carmen Morales	
<p>Descripción:</p> <p>Se realiza el diseño de diagrama de secuencia</p>	

Tabla 6. Tarea 2.1 Historia de Usuario 2

Tarea	
Número: 2.2	Número historia: 2
Nombre tarea: Creación de menú y proyección del mismo en la mesa de billar	
Tipo de tarea: Desarrollo	Puntos Estimados: 1
Fecha inicio:04-01-2012	Fecha fin:10-01-2012
Programador responsable: Carmen Morales	
<p>Descripción: Se dibuja un menú con opciones establecidas, posteriormente se determina en qué coordenada se encuentra cada opción dentro de la mesa de billar y finalmente es proyectada sobre la misma.</p>	

Tabla 7. Tarea 2.2 Historia de Usuario 2

Tarea	
Número: 2.3	Número historia: 2
Nombre tarea: Determinación de opción seleccionada	
Tipo de tarea: Desarrollo	Puntos Estimados: 1
Fecha inicio:11-01-2012	Fecha fin:17-01-2012
Programador responsable: Carmen Morales	
Descripción: Al tenerse las opciones proyectadas sobre la mesa, se elige con la mano una de ellas y se realiza una comparación con una máscara almacenada previamente luego se indica cual fue la alternativa escogida.	

Tabla 8. Tarea 2.3 Historia de Usuario 2

Tarea	
Número: 2.4	Número historia: 2
Nombre tarea: Proyección de función seleccionada y modo de espera de menú	
Tipo de tarea: Diseño	Puntos Estimados: 1
Fecha inicio:18-01-2011	Fecha fin:25-01-2011
Programador Responsable: Carmen Morales	
Descripción: enviar función solicitada y mantener activo el algoritmo hasta que se desee finalizar utilización de menú.	

Tabla 9. Tarea 2.4 Historia de Usuario 2