Universidad Nacional Autónoma de México

FACULTAD DE CIENCIAS

Software interactivo para enseñanza en Matemáticas

REPORTE DE ACTIVIDAD DOCENTE

QUE PARA OBTENER EL TÍTULO DE:
LICENCIADO EN CIENCIAS DE LA COMPUTACIÓN

P R E S E N T A :
GILDARDO BAUTISTA GARCÍA CANO

T U T O R A :
MAT. MÓNICA LEÑERO PADIERNA

2009

VNIVERSIDAD NACIONAL
AVTNOMA DE
MEXICO

FACULTAD DE CIENCIAS
Secretaría General
División de Estudios Profesionales

Votos Aprobatorios

ACT. MAURICIO AGUILAR GONZÁLEZ
Jefe de la División de Estudios Profesionales
Facultad de Ciencias
P r e s e n t e

Por este medio hacemos de su conocimiento que hemos revisado el trabajo escrito titulado:

Sofware interactivo para enseñanza en Matemáticas

realizado por **Bautista García Cano Gildardo** con número de cuenta **0-9528098-9** quien ha decidido titularse mediante la opción de **actividad de apoyo a la docencia** en la licenciatura en **Ciencias de la Computación**. Dicho trabajo cuenta con nuestro voto aprobatorio.

Propietario	Dr. Carlos Hernández Garciadiego	
Propietario	M. en C. Patricia Martínez Falcón	
Propietario	Mat. Mónica Leñero Padierna	
Tutora		
Suplente	Dr. José de Jesús Galaviz Casas	
Suplente	Dra. Elisa Viso Gurovich	

Atentamente,
"POR MI RAZA HABLARÁ EL ESPÍRITU "
Ciudad Universitaria, D. F., a 27 de marzo de 2009
EL COORDINADOR DEL COMITÉ ACADÉMICO DE LA LICENCIATURA EN CIENCIAS DE LA COMPUTACIÓN

FACULTAD DE CIENCIAS
CONSEJO DEPARTAMENTAL
DE
MATEMATICAS

DRA. ELISA VISO GUROVICH

Señor sinodal: antes de firmar este documento, solicite al estudiante que le muestre la versión digital de su trabajo y verifique que la misma incluya todas las observaciones y correcciones que usted hizo sobre el mismo.

A mis padres que son mi ejemplo de vida.

Agradecimientos

A mis padres: Magdalena y Francisco, por apoyarme siempre en todos los aspectos, por todos sus consejos, por siempre estar conmigo y por ser como son, gracias.

A mis hermanos: Maricela y Fernando, a quienes les tocó ser los hermanos mayores y vivir cosas antes que yo, para que mi camino fuera más sencillo.

A mis abuelos Rodolfo y Consuelo, a mi tía Rosaura por darme un gran primo Sergio y a mi tía Graciela quien invirtió en clases de inglés para que su sobrino se superara. Muchas gracias por todo.

A mi tutora, jefa y amiga Mónica Leñero, quien me brindó la oportunidad de trabajar con ella y quien nunca se cansó de recordarme lo importante que es concluir esta etapa. Gracias por tus consejos, tiempo y confianza.

A mi consentida Adriana Ramírez, gracias por estar conmigo durante todos estos años. Por apoyarme en todo momento, por impulsarme a seguir estudiando y conquistando lugares lejanos. Espero sigamos disfrutando la vida juntos.

A José Galaviz por todos los conocimientos que me trasmitió en cada materia que tomé con él. Gracias por darme la oportunidad de dar mi primer clase como ayudante en la facultad y por brindarme tu amistad.

A Elisa Viso por los cursos de algoritmos y teoría de la computación, ya que fueron las mejores materias que tomé durante la carrera. Gracias también por los talleres extracurriculares como el de Latex y Java.

A Paty, Gaby y Marina porque fue una muy grata experiencia trabajar con ustedes durante el desarrollo de las aplicaciones de Matechavos. Espero poder seguir colaborando con ustedes en el futuro.

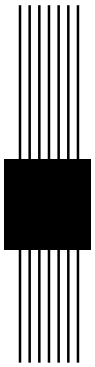
A Carlos Hernández por tomarse el tiempo de revisar este trabajo.

A Vicky y Crevel por compartir el deseo de estudiar y divertirse. Gracias por todos los momentos que pasamos juntos en la fac y los viajes como el de Oaxaca.

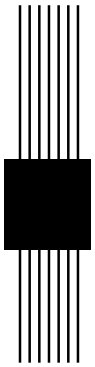
A Mitzi y Polo por no formar parte de mi vida académica, pero si social y deportiva aunque ya solo sea en box, digo Xbox.

Gracias a todos, este trabajo es de ustedes.

Índice general



Índice de figuras



Introducción 1

1.1 Marco teórico de apoyo a la docencia

Sin duda, uno de los problemas más importantes que enfrentan la difusión y la enseñanza formal de las matemáticas es la idea generalizada de que no son fáciles y de que son una disciplina basada en fórmulas aprendidas de memoria, lo cual genera una actitud de rechazo hacia esta materia. Los motivos que apoyan esta idea errónea pueden ser muy diversos: planas y planas de ejercicios similares; tareas matemáticas impuestas como castigos escolares; temas que no se vinculan con la vida cotidiana, lo que se traduce en falta de interés por la materia; el énfasis excesivo que suele darse a la memorización de resultados. Este último punto es muy importante ya que al pedir al estudiante que memorice lo estamos enseñando a hacer las cosas de manera automática, lo que lo aparta de comprender el porqué se hace de esa manera (concepto) y poder generar soluciones a problemas similares.

Por ejemplo, si a un niño se le enseña a multiplicar repitiendo las tablas una y otra vez hasta memorizarlas, será capaz de resolver su plana de multiplicaciones pero no comprenderá en qué consiste la multiplicación y le costará trabajo trasladarla a otros ámbitos, como por ejemplo a las fracciones. Si le enseñamos a través de situaciones que lo desafíen a encontrar procedimientos para resolver problemas y a través de esos problemas comprende

en qué consiste la multiplicación, será capaz de transferir ese aprendizaje. Ante esta visión de las matemáticas, el área de docencia ha hecho uso de las computadoras como herramientas de apoyo en el proceso de enseñanza-aprendizaje, utilizando recursos multimedia, como son: imágenes, videos, audio e incluso juegos. Uno de los beneficios de usar estas herramientas es que podemos implementar aplicaciones atractivas (con un buen diseño gráfico) e interactivas (retroalimentando las acciones del usuario) que llamen la atención del usuario y que representen situaciones en las que se vea obligado a construir sus propias soluciones, desarrollando (construyendo) así el conocimiento.

Otra ventaja es que se pueden usar programas que respondan a las acciones que realice el usuario, retro-alimentando de manera visual si las respuestas a un problema planteado son correctas. Esto es de mucha ayuda sobre todo con alumnos de primaria, en quienes lo visual capta la atención. Este paradigma favorece además la seguridad de los niños al no necesitar a un adulto que les diga si lo que hicieron está bien o no y permite que el usuario pruebe sus propuestas de solución tantas veces como lo desee, pues el programa interactúa directamente con él.

Sin embargo la mayoría de los profesores no saben cómo desarollar estas aplicaciones. Gracias a su experiencia frente al grupo tienen muchas ideas acerca de cómo presentar el material a los niños, acerca de los conceptos, la dinámica, actividades previas y para reforzar, etc. Saben lo que les gustaría que un programa de apoyo tuviera, pero no saben cómo programarlo. Lo que pueden hacer, dado que la carga de trabajo normalmente les impide tomar todos los cursos de capacitación que necesitarían para programar actividades interactivas, es buscar en Internet algo que les ayude en su labor docente. Lamentablemente la mayoría de las aplicaciones que se encuentran son muy caras, están en Inglés o Francés y las que hay en Español no reflejan las situaciones ni el lenguaje cotidiano de nuestros niños (por ejemplo, las que están desarrolladas en España, presentan algunos problemas con el lenguaje y con la notación decimal). En un porcentaje muy alto, el material que se consigue son ejercitadores (programas que a través de la repetición favorecen sólo la memorización) o material sin apoyo pedagógico. En resumen, hay pocas aplicaciones que apoyan la enseñanza que puedan utilizarse en la educación básica y media.

Las aplicaciones presentadas en este trabajo pretenden ser una herramienta para el apoyo de la enseñanza de las Matemáticas en educación básica

y media. Fueron desarrolladas como parte de PUEMAC (Proyecto Universitario de Enseñanza de las Matemáticas Asistida por Computadora, http://puemac.matem.unam.mx). Este es un proyecto del Instituto de Matemáticas de la UNAM[1], en colaboración con varias dependencias de la misma Universidad (DGSCA[2], Fac. de Ciencias[3], IIMAS[4]) y que cuenta con el apoyo de diversas instituciones preocupadas por la enseñanza y divulgación de las Ciencias (AMC[5], ILCE[6], CONACYT[7], SMM[8]). El grupo de trabajo de PUEMAC está formado por investigadores y profesores de matemáticas y ciencias de la computación; pedagogas especialistas en educación matemática y trabajo con niños; diseñadores gráficos y programadores de alto nivel. Uno de los objetivos del proyecto es dotar a la comunidad con material para apoyo a la docencia en el área de Matemáticas, en todos los niveles educativos.

Las secciones de PUEMAC están orientadas por la premisa de que las computadoras son herramientas que pueden apoyar la actividad de enseñanza, en tanto sean utilizadas con una clara intención didática por parte del docente. La sección dedicada a los niños (Matechavos) ha sido cuidadosamente diseñada por el equipo de asesoría pedagógica, personal de Cómputo para niños de DGSCA. Las actividades de esta sección se han desarrollado con el enfoque de la didáctica constructivista. Se plantean al niño, en el contexto de un juego, problemas en cuya solución están involucrados conceptos matemáticos. Además, se permiten varios procedimientos en la solución. La idea es que al dar las soluciones generen estrategias y, sin que se percaten de ello, los niños se apropien de los conceptos y conocimientos matemáticos.

Matechavos tiene el propósito de presentar a los docentes y niños una propuesta de apoyo a la enseñanza de las matemáticas en la escuela primaria, que complementa los nuevos planes y programas de estudio y se fundamenta en las más recientes investigaciones en Didáctica de las matemáticas, llevadas a cabo en los mejores centros de investigación educativa de nuestro país y en el extranjero. No pretende ser una propuesta que reemplace a la escuela o al docente, sino una herramienta de trabajo para que los maestros puedan

[1] www.matem.unam.mx
[2] http://www.dgsca.unam.mx
[3] http://www.fciencias.unam.mx
[4] http://www.iimas.unam.mx
[5] http://www.amc.unam.mx
[6] http://www.ilce.edu.mx
[7] http://www.conacyt.mx
[8] http://www.smm.org.mx

realizar su labor contando con materiales innovadores.

Algunos de los requerimientos de Matechavos, y de PUEMAC en general, eran mantener el contenido disponible a todo público de manera gratuita y presentar aplicaciones atractivas e intuitivas que llamaran la atención de los niños, sin tener que saturarlas de animaciones u objetos que pudieran distraerlos del objetivo principal. La manera más viable de cumplir el primer punto fue hacer disponible nuestras aplicaciones por internet, por el gran alcance que tiene este medio de comunicación y dado que cada día más y más gente tiene acceso a él. Para cumplir el segundo punto fue muy importante la comunicación constante con el equipo pedagógico y de diseño, ya que muchas veces la presentación y los requerimientos funcionales iniciales eran modificados. Cambios en el tipo de fuente o tamaño de los textos, en la posición o colores de los objetos y en las acciones de retroalimentación surgían al hacer pruebas en las escuelas. Para la implementación de las actividades se buscó una tecnología que nos permitiera desarrollar aplicaciones que se pudieran mostrar en un navegador, se pudieran ejecutar en la mayoría de los sistemas operativos (o al menos los más usados) y que los programas que tuvieran que instalar los usuarios fueran mínimos. Como posibles tecnologías teníamos *Java Applets* y *Flash*. A diferencia de *Flash*, *Java* es un lenguaje de programación de alto nivel orientado a objetos, lo cual nos facilita la implementación y manejo de estructuras de datos y nos permite controlar la lógica de cada aplicación de manera precisa. Además cuenta con un api (conjunto de bibliotecas) bastante robusto para el desarrollo de interfaces gráficas y para el manejo de eventos, tanto del teclado como del ratón. Por todo lo anterior se tomó la decisión de utilizar Java.

Aplicaciones Desarrolladas

2.1 Applets de Java

Un *applet* de Java, o *applet*, es una aplicación escrita en *Java* que un navegador, que cuente con un *plugin*[1] adecuado puede descargar y ejecutar. Para agregar un applet a una página Web se agrega una etiqueta de HTML que indica al navegador de dónde descargarlo; en cuanto termina la descarga el applet se ejecuta en la máquina cliente. Como en el caso de las imágenes, los applets ocupan un área rectangular dentro de la página y además permiten interactuar con el usuario, lo que los hace perfectos para el desarrollo de juegos en línea.

La posibilidad de agregar contenido dinámico en un página Web es una de las principales ventajas de los applets y esta característica ayudó en gran medida a la aceptación del lenguaje *Java* en sus inicios. Una de sus restricciones es que dado que son aplicaciones descargables de Internet, se ejecutan en un ambiente restringido y no se les permite realizar lecturas o escrituras en la máquina local, por cuestiones de seguridad.

[1] Es una aplicación que interactúa con otra, para aportarle una función o utilidad específica.

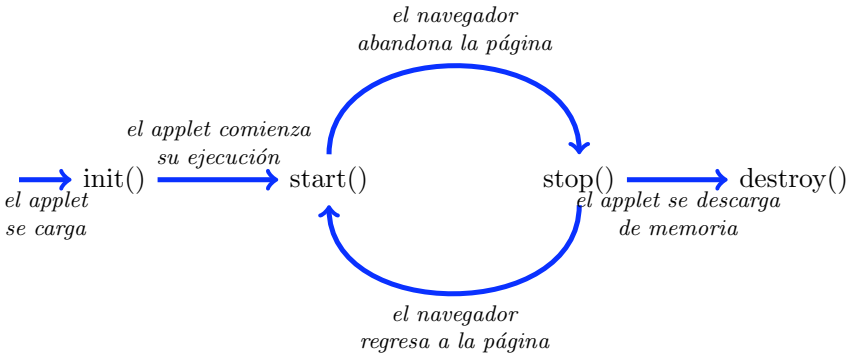


Figura 2.1: Ciclo de vida de un *applet*

2.1.1 El ciclo de vida de un applet

A diferencia de una aplicación normal escrita en *Java*, la cual comienza su ejecución a partir de su método *main* y crea su propia interfaz gráfica, un *applet* es un componente que espera ser usado dentro de la interfaz gráfica de alguien más, es decir no puede ejecutarse de manera independiente. Un navegador interactúa con un *applet* a través de los métodos: *init, start, stop* y *destroy*, los cuales controlan su ciclo de vida (ver figura 2.1).

init. Se llama sólo una vez, cuando el *applet* es cargado. Se evalúan parámetros, se crea la interfaz de usuario y se cargan recursos (imágenes, audio).

start. Este método se llama automáticamente después de *init* y cuando regresamos a la página que contiene el *applet*, después de visitar otra página. Aquí comienza la ejecución del *applet*. Si es necesario crear algún hilo (*thread*) éste es el lugar indicado.

stop. Si el usuario abandona la página en la que está el *applet* o cierra el navegador, se llama a este método. Esto nos permite detener hilos o código que estén en ejecución.

destroy. Es llamado antes de que el *applet* sea descargado de memoria.

Un *applet* puede contener los mismos componentes gráficos que cualquier otra aplicación en Java, se puede dibujar sobreescribiendo su método *paint* y puede responder a eventos si se implementan las interfaces adecuadas.

En general, no hay restricciones para hacer un *applet* tan complejo como se desee, excepto quizá la velocidad de descarga, ya que cada clase que utiliza es descargada antes de comenzar su ejecución. Una vez que todas las clases son descargadas el *applet* se ejecuta usando los recursos de la máquina cliente. El código base de un *applet* se muestra en el listado 2.1y se presenta sólo como referencia.

```
import javax.swing.*;
import java.awt.*;

public class MiApplet extends JApplet {

    public void init() {
        System.out.println("Creando Applet");
    }

    public void start() {
        System.out.println("Iniciando Applet");
    }

    public void paint(Graphics g) {
        g.drawString("Java Applet", 5, 15);
    }

    public void stop() {
        System.out.println("Deteniendo Applet");
    }

    public void destroy() {
        System.out.println("Destruyendo Applet");
    }
}
```

Listado 2.1: Estructura basica de un *applet*

A continuación se presentan algunas de las aplicaciones desarrolladas para el proyecto PUEMAC; todas se encuentran en la sección Matechavos.

Figura 2.2: Cuadrados mágicos

2.2 Cuadrados Mágicos

El juego consiste en acomodar 9 de 12 números dados en un tablero de tres renglones por tres columnas, de tal manera que la suma en sentido vertical, horizontal y diagonal dé siempre el mismo resultado. Con este juego se pretende que los niños practiquen operaciones aritméticas básicas.

2.2.1 Objetivos

Identificar distintas combinaciones de números que, al sumarse entre sí, dan el mismo resultado. Desarrollar estrategias que permitan acomodar los números para que las sumas den el resultado requerido.

2.2.2 Requisitos pedagógicos y de diseño

Visualmente la aplicación está dividida verticalmente en dos secciones principales: área de juego (lado izquierdo) y área de configuración (lado derecho). En el centro del área de juego aparece un tablero formado por nueve cuadrados y alrededor de él se distribuyen de manera aleatoria, pero sin encimarse, doce números consecutivos, los cuales tienen que ser acomodados en el tablero. Cada número está encerrado en un círculo y utilizando el ratón pueden ser seleccionados y arrastrados dentro del área de juego. En caso de que un círculo sea soltado sobre alguno de los cuadrados vacíos, y quede completamente contenido, tomará la forma del cuadrado y cambiará de color a uno predefinido. Si es necesario retirar un número previamente colocado sobre el tablero basta con seleccionarlo y arrastrarlo fuera, volviendo a ser un círculo. Un círculo es seleccionado dándole un clic con el botón izquierdo del ratón y se arrastra manteniendo presionado el botón.

La suma esperada se indica arriba del tablero y cambia en cada juego. Cuando se logran acomodar los cuadrados de tal manera que las sumas verticales, horizontales y diagonales corresponden a la indicada, a manera de retroalimentación se muestra un mensaje animado, que indica al jugador que ha acomodado los números correctamente. No hay restricción de tiempo para completar el cuadrado ya que el objetivo principal es practicar las sumas. Para comenzar un juego nuevo hay un botón debajo del tablero que cambia la suma esperada y reacomoda los números.

Este juego consta de tres niveles de dificultad los cuales son accesibles desde el área de configuración. El nivel fácil comienza con tres de los nueve números ya acomodados en el tablero, con la restricción de que no aparezcan dos en la misma columna, renglón o diagonal, y pintados de color verde para distinguir el nivel. Las sumas posibles son 12, 15, 18, ...39. El nivel intermedio comienza sólo con un número dentro de un cuadrado azul acomodado en el tablero, pero no puede ser el del centro. En este nivel se da la posibilidad de pedir hasta dos pistas por juego, en cuyo caso se muestra parpadeando en un cuadrado un número de la solución. Las sumas posibles son 24, 27, ...66. El nivel difícil inicia con el tablero vacío y también se tiene la posibilidad de pedir dos pistas. Las sumas posibles son 24, 27, ...78. Para distinguir el nivel en que se está jugando se muestra un letrero que indica el nivel actual, así como dos botones que permiten cambiar a los otros niveles. Por omisión el juego comienza en el nivel fácil.

Figura 2.3: Cuadrados mágicos

Para mostrar las instrucciones e información de los colaboradores en este juego (idea original, diseñador y programador) hay dos botones en el área de configuración, que abrirán sendas ventanas.

Como último requerimiento el applet se debe desplegar de manera completa en un monitor cuya resolución mínima sea de 800x600, tomando en cuenta el espacio de la ventana del navegador, el encabezado del proyecto (PUEMAC) y el índice de matechavos, que se muestran en todas las páginas de dicha sección. Al final el juego se ve como en las figuras 2.2 y 2.3.

2.2.3 Implementación

Antes de comenzar con la descripción de la implementación describiremos brevemente algunos conceptos y técnicas usadas en el desarrollo de juegos de video.

La parte más importante en cualquier juego es el algoritmo de animación,

ya que de él depende que el juego sea fluido y que la secuencia de acciones se vea lo mejor posible, independientemente de la plataforma en que se ejecute. Generalmente el algoritmo de animación consta de tres etapas: *actualiza*, *muestra* y *espera*, las cuales se ejecutan de manera iterativa hasta que termina la aplicación. Este ciclo se conoce como el ciclo de animación y se muestra en el diagrama de la figura 2.4.

En la etapa *actualiza* se puede modificar el estado de los objetos, revisar su interacción (por ejemplo colisiones) y llevar el control del juego (marcador). Algunos objetos pueden cambiar de estado como respuesta a las acciones del usuario, dando así interactividad al juego. La etapa *muestra* se encarga de dibujar todos los objetos que tengan representación gráfica en un *buffer* y cuando el *buffer* está listo se muestra en la pantalla. La técnica de usar un *buffer* ayuda a eliminar el parpadeo que se puede generar al ir pintando uno a uno los objetos en la pantalla. La última etapa, *espera*, es una pequeña pausa que nos ayuda a controlar el número de cuadros por segundo que se muestran; de esta manera tenemos una secuencia de animación que no depende de la computadora en que se ejecute.

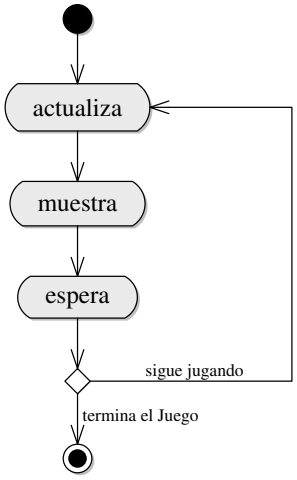


Figura 2.4: Ciclo de animación

Un concepto bastante usado cuando hablamos de juegos o animaciones es el de *Sprite*. Un *sprite* es una figura u objeto en el juego que puede moverse

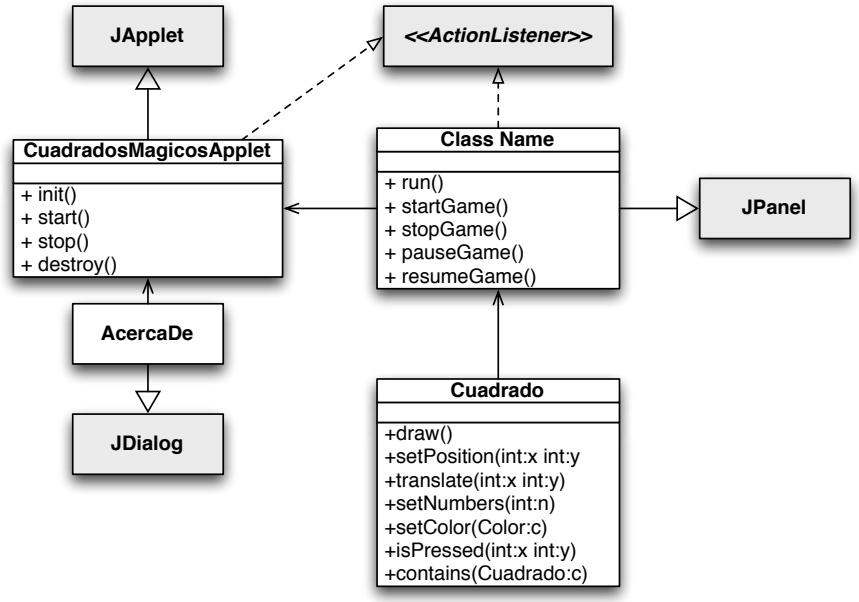


Figura 2.5: Diagrama de clases del *applet* Cuadrados Mágicos

independientemente de otros, puede ser un texto o cualquier objeto que pueda pintarse. Como cualquier otro objeto un *sprite* tiene atributos que definen su estado y métodos que definen su comportamiento. La mayoría de las implementaciones toman como atributos la posición, tamaño, estado, una imagen (para la representación visual de cada uno), prioridad (que nos indica el orden en que deben pintarse en caso de que se intersecten) y visibilidad (para saber si se pintan o no). Entre los métodos más comunes tenemos el de pintar y actualizar.

Ahora describiremos las responsabilidades y la manera en que interactúan las clases que componen el juego de Cuadrados Mágicos. En la figura 2.5 se muestra el diagrama de clases para este juego.

La clase Cuadrado representa un *sprite* que puede moverse libremente sobre el área de juego; su posición puede modificarse de dos maneras: dando las nuevas coordenadas explícitamente usando el método setPosition(int x, int y) o con el método translate(int x, int y) que incrementa la posición actual en x

y *y* pixeles. Cada ejemplar de la clase Cuadrado representa un número que es el que tomamos en cuenta al hacer las sumas en el juego y debe ser un entero positivo o -1 en caso de ser un cuadrado vacío. Los cuadrados vacíos se usan para representar en el tablero dónde se pueden colocar los demás cuadrados, en la figura 2.2 son los de color blanco. El color de los cuadrados depende del nivel del juego y de si aparecen fijos en el tablero, atributo que puede modificarse usando el método setColor(Color c). Dados los requerimientos necesitamos distinguir los cuadrados que están en el tablero de los que están fuera de él y para esto el equipo pedagógico pidió que fuera del tablero los números se mostraran en círculos. Tal vez el nombre de la clase ya no sea el más adecuado, pero este cambio surgió después de que la aplicación fue usada en varias escuelas. Para saber cuándo se arrastra un círculo y se suelta sobre un cuadrado del tablero, se tienen los métodos isPressed(int x, int y) y contains(Cuadrado c). El primero indica si la coordenada (x,y) donde se dio clic con el ratón, pertenece o no al cuadrado y el segundo dice si un cuadrado está dentro de otro. En la implementación, para decidir si un cuadrado contiene a otro se da un pequeño margen de error (delta), que consiste en unos cuantos pixeles que pudieran faltar para que estuviera completamente contenido, pues dado que el jugador podría no ser un experto con el ratón es mejor permitir este margen de error. El código de este método se muestra en el listado 2.2.

```
public boolean contains(Cuadrado c){
    int delta = 4;
    return( x < (c.getX() + delta) &&
    (c.getX()+c.getWidth()-delta) < (x+getWidth()) &&
    y < (c.getY() + delta) &&
    (c.getY()+c.getHeight()-delta) < (y+getHeight()));
}
```

Listado 2.2: Cuadro.contains(Cuadrado c)

Por último tenemos los métodos setImage(Image img), que asigna la imagen de círculo o cuadrado y draw(Graphics g) que se encarga de pintar el cuadrado en el *buffer*. Primero pinta la imagen que tiene asignada y después, usando el color que tenga definido, pinta el número que representa si es distinto de -1.

La clase CuadradoMagicoPanel representa el área de juego donde estará el tablero y donde se podrán mover los cuadrados. En esta clase se cargan

las imágenes que se pueden asignar a los cuadrados y las imágenes que se muestran cuando se gana el juego.

Durante toda la ejecución del *applet* sólo se crean 21 ejemplares de la clase Cuadrado, nueve permanecen fijos para formar el tablero y los otros 12 serán los que se podrán mover en el panel. Estos 12 cuadrados estarán dentro de una lista y cuando sea necesario pintarlos será en orden inverso a la lista, es decir comenzaremos pintando el último. Cuando se dé un clic dentro del panel se preguntará a cada cuadrado, en el orden en que aparece en la lista, si fue seleccionado. El primero que responda de manera afirmativa será trasladado a la primera posición de la lista y de esta manera, a la hora de arrastarlo y volver a pintar los cuadrados, tendremos el efecto de que el que seleccionamos pasa sobre los demás.

2.3 La Pulga y las Trampas

El Juego consiste en hacer saltar a una pulga sobre un camino, evitando caer en las trampas que se encuentran colocadas en él. También se puede elegir colocar las trampas e intentar hace fallar a la pulga. Esta aplicación puede jugarse contra la computadora o con alguien más.

2.3.1 Objetivo

El camino es representado por una recta numérica de varias unidades y cuando se recorre, todos los saltos son del mismo tamaño, lo cual permite identificar y practicar las series numéricas y la multiplicación. Surgirán las nociones de múltiplos comunes, mínimo común múltiplo y máximo común divisor.

2.3.2 Requisitos pedagógicos y de diseño

Al iniciar debe seleccionarse el nivel, si hay uno o dos jugadores y el papel que se quiere desempeñar: ser la pulga o ser quien pone las trampas. Hay tres niveles de dificultad que determinan el tamaño de la recta y el número de trampas. En el primer nivel la recta es de 20 unidades y se coloca una trampa, en el segundo nivel la recta mide 30 unidades y se colocan dos trampas, en el tercer nivel la recta tiene 40 unidades y hay tres trampas.

La recta siempre debe mostrar todas la unidades, dependiendo del nivel, y las trampas se colocan sobre la recta. Cuando la computadora coloca las

Figura 2.6: La pulga y las trampas

trampas debe hacerlo después del 9 para forzar al jugador a usar el concepto de múltiplos. Si se elige poner las trampas, éstas aparecen en la esquina superior izquierda y se deben arrastrar con el ratón hasta la recta. Cuando se coloca una trampa sobre la recta se muestra visualmente en qué unidad quedará colocada.

La pulga sólo puede avanzar sobre la recta saltando, por lo que el jugador deberá seleccionar la longitud del salto, que puede ser de 2 a 9 unidades. Una vez seleccionada la longitud del salto se debe dar clic sobre el botón "¡A saltar!", para que la pulga comience a recorrer la recta. Si logra llegar al final o si cae en una trampa se muestra una animación como retroalimentación al usuario.

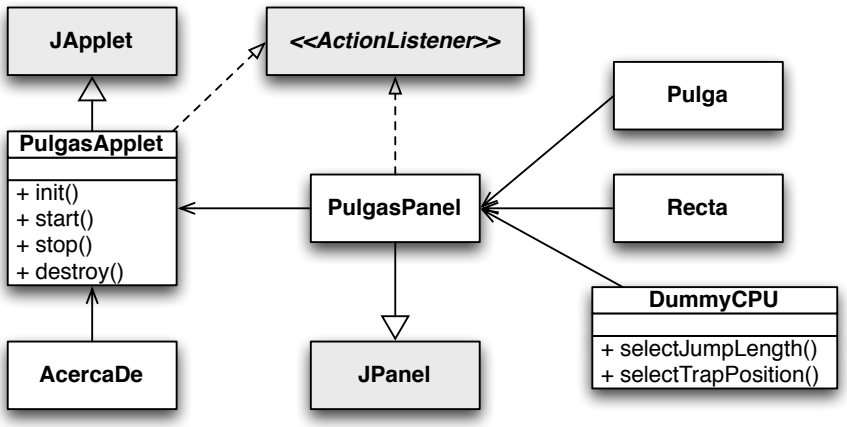


Figura 2.7: Diagrama de clases del *applet* La pulga y las trampas

2.3.3 Implementación

Dados los requerimientos se identificaron las clases mostradas en el diagrama UML que se muestra en la figura 2.7. En este juego varias de las clases tienen una implementación muy breve, pero se consideraron indispensables para mantener una organización más clara.

Comenzaremos describiendo la clase Recta, la cual tiene como objetivo mostrar de manera gráfica la representación de la recta correspondiente a cada nivel, junto con su graduación, esto con la finalidad de que el jugador reciba una retroalimentación visual de la longitud de cada salto. Internamente se implementó con un arreglo cuyo tamaño depende del nivel, y cada elemento del arreglo es Verdadero o Falso dependiendo de si en la posición existe o no una trampa.

La clase Pulga corresponde al personaje principal de este juego. La posición inicial sobre la recta se representa con un entero y comienza siempre en cero. Cuando la pulga comienza a saltar esta posición se incrementa dependiendo de la longitud del salto. También tiene asociado un estado que indica si se encuentra en el inicio del juego, si está saltando, si cayó en una trampa o si logró recorrer toda la recta.

En la clase DummyCPU se implementa la habilidad de la máquina para colocar trampas o seleccionar la longitud del salto. Dada la edad de los niños a quienes va dirigida la aplicación, se decidió que el algoritmo no sea muy inteligente, pues esto motivará al niño a seguir jugando y de esa manera alcanzar los objetivos pedagógicos del juego.

PulgasPanel es la encargada de organizar la interacción de ejemplares de las clases antes descritas y de manejar los eventos que recibe por parte de la clase principal (PulgasApplet). Por ejemplo cuando la pulga comienza a saltar, se encarga de revisar en qué posición de la recta cae y verifica dentro del arreglo de el ejemplar Recta si hay una trampa o no.

Finalmente la clase PulgasApplet es la responsable del manejo de la configuración (nivel, número de jugadores y si se juega como pulga o poniendo las trampas) y el control del juego (botones juego nuevo y comenzar a saltar). Cuando el usuario usa alguno de estos botones se genera un evento que se envía a la clase PulgasPanel, para que realice la acción asociada a dicho evento.

2.4 A Romper Globos

Este es el tradicional juego de la feria del mismo nombre. El objetivo es romper todos los globos que aparecen colocados sobre una recta dado un número fijo de dardos.

La adaptación consiste en considerar una recta cuya longitud sea de una o dos unidades según el nivel y colocar los globos sobre la recta, de manera que estén exactamente sobre fracciones específicas. La recta no tiene indicadores numéricos.

Para jugar el niño debe elegir un nivel, oprimir el botón globos y escoger la fracción hacia donde se lanzará el dardo.

2.4.1 Objetivo

Identificar las fracciones en la recta, su orden y sus equivalencias.

2.4.2 Requisitos pedagógicos y de diseño

La recta mide una unidad en el primer nivel y dos en los demás, sin mostrar indicadores numéricos. En todos los niveles hay que romper cinco

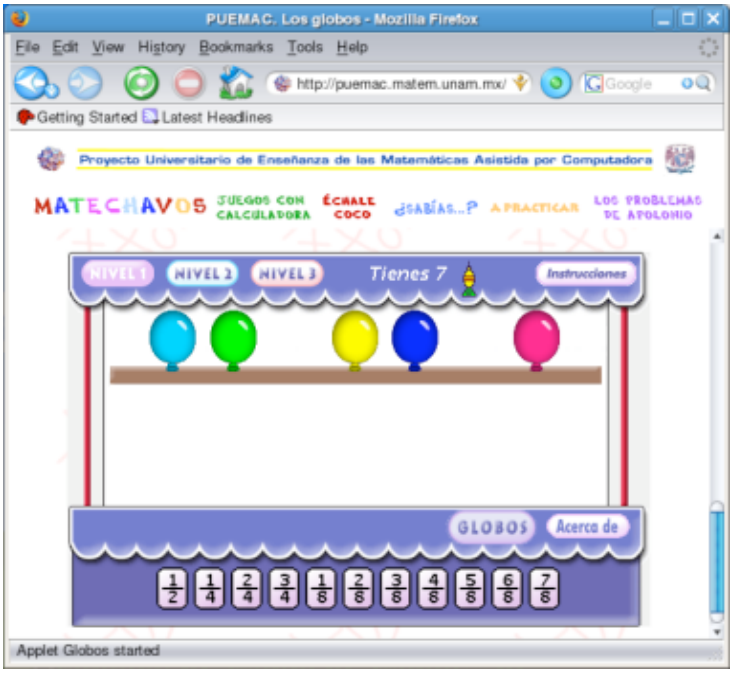


Figura 2.8: Globos

globos. El lugar en donde se quiere lanzar el dardo se escoge seleccionando una fracción de la parte inferior de la aplicación. En el primer nivel se manejan medios, cuartos y octavos y se cuenta con siete dardos. En el segundo nivel se manejan los mismos denominadores y se tienen nueve dardos. En el tercer nivel sólo pueden usarse medios, tercios, cuartos y sextos, con nueve dardos. Para comenzar el juego los globos deben moverse desde la parte inferior de la aplicación hasta colocarse sobre la recta, dando el efecto de que se están elevando. Cuando se lanza un dardo también debe mostrarse cómo sale disparado y una vez que llega a la recta permanece visible e indicando qué fracción representa, aún cuando se trate de fracciones equivalentes. En caso de que le atine a un globo, éste debe romperse.

Cuando se logra romper todos los globos se premia con un dardo adicional en el siguiente juego y si sobraron dardos también se acumulan.

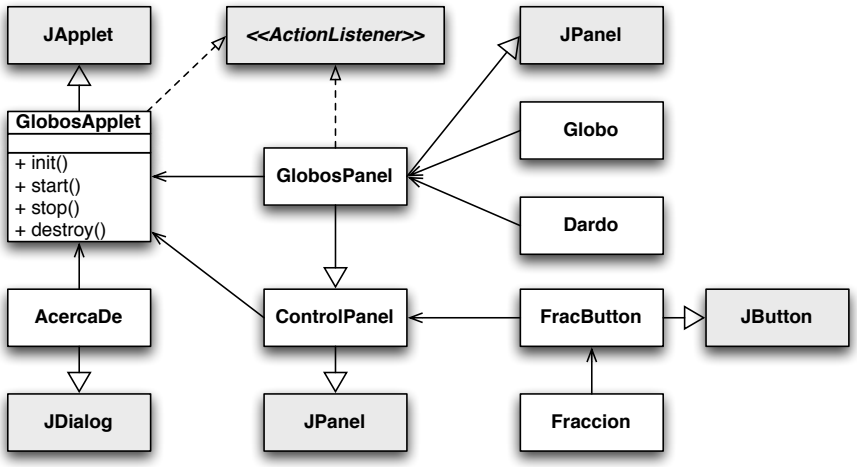


Figura 2.9: Diagrama de clases del *applet* Globos

2.4.3 Implementación

De los requerimientos mencionados podemos concluir que necesitamos representar a los globos y a los dardos como objetos de tipo *sprite*, ya que tendrán movimiento durante en el juego. Las clases correspondientes se muestran en la figura 2.9.

Cuando se inicia el juego los globos se elevan hasta llegar a puntos sobre una recta, donde se detienen; dichos puntos representan fracciones de la recta. Para simular este movimiento cada globo tiene una posición inicial y una final, que definen la trayectoria que debe seguir al elevarse; esta trayectoria se recorre en un tiempo fijo definido dentro de la misma clase. Se mantienen dos estados: uno indica si el globo está en movimiento o no y el otro indica si el globo está inflado o ya ha sido roto. Además guardamos la fracción sobre la cual queda detenido el globo, para posteriormente verificar si algún dardo se dispara hacia dicha fracción.

Cuando los globos ya están sobre la recta es tiempo de romperlos. Para esto utilizamos dardos que también siguen una trayectoria hacia fracciones de la recta. En este caso mantenemos los mismos atributos que los globos, eliminando el estado que indica si el globo estaba inflado.

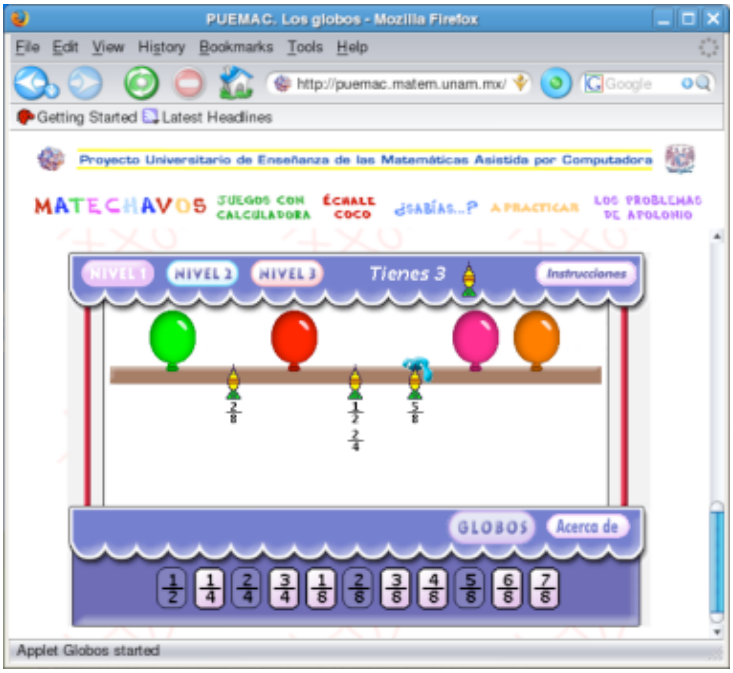


Figura 2.10: Globos

El usuario sólo puede lanzar dardos en fracciones válidas de la recta. Para realizar los tiros tenemos botones que se restringen a estas fracciones. Los botones que definimos fueron con la única finalidad de poder mostrar las fracciones de la manera tradicional ($\frac{p}{q}$). Todos los botones se encuentran dentro de un panel de control (ControlPanel), que se encarga de mostrar los botones de manera distribuida a lo largo de la aplicación.

GlobosPanel se encarga de crear y controlar la interacción de los elementos mencionados. Cuando inicia la aplicación se crea un arreglo de cinco globos y otro de nueve dardos. Cuando el usuario oprime un botón se le asigna la fracción correspondiente al primer dardo y en ese momento se calculan las posiciones iniciales y finales de dicho dardo y se cambia al estado de movimiento. Inmediatamente después comienza la animación. Cuando el dardo termina el recorrido de su trayectoria se regresa al estado inmóvil y se verifica sobre el arreglo de globos si alguno coincide en la fracción de dicho dardo, lo cual indicaría que le atinó a un globo.

2.5 Batalla Naval

Este juego está basado en el clásico juego de Submarino donde se tiene una cuadrícula de 10x10 y varios barcos de distintos tamaños, que están acomodados estratégicamente en la cuadrícula. Cada cuadrado dentro de la cuadrícula está identificado por una pareja de números (x, y), cada barco puede estar colocado horizontal o verticalmente y ocupa un número consecutivo de cuadros dependiendo de su tamaño. El juego consiste en tratar de encontrar todos los barcos que se encuentran ocultos bajo el agua lanzando bombas en distintas coordenadas. Un barco se considera hundido si se ha colocado una bomba en cada cuadrado que ocupa. El juego termina cuando se hunden todos los barcos o se acaban las bombas.

2.5.1 Objetivo

El propósito didáctico de este juego es practicar la ubicación de coordenadas en el plano cartesiano.

2.5.2 Requisitos pedagógicos y de diseño

El juego dispone de tres niveles de dificultad. En el nivel fácil se trabaja con el cuadrante positivo en el rango 0-10 en ambos ejes. En el nivel intermedio se trabaja nuevamente con el cuadrante positivo pero el rango varía entre 0-10 y 15-25. En el nivel difícil se trabaja con cualquiera de los cuatro cuadrantes y con rangos similares a los del nivel intermedio.

El área de juego es una cuadrícula de 10x10 que se encuentra numerada en los ejes X y Y. Dentro de este cuadrado se ocultan cuatro barcos de tamaños 2, 3, 4 y 5 unidades, de manera que no queden encimados. Los barcos se colocan vertical u horizontalmente en el tablero (no en diagonal). El jugador cuenta con 45 bombas que puede utilizar seleccionando una coordenada en X y otra en Y.

En este caso el juego consta de tres áreas principales: el área de configuración, el área de juego y el área de controles.

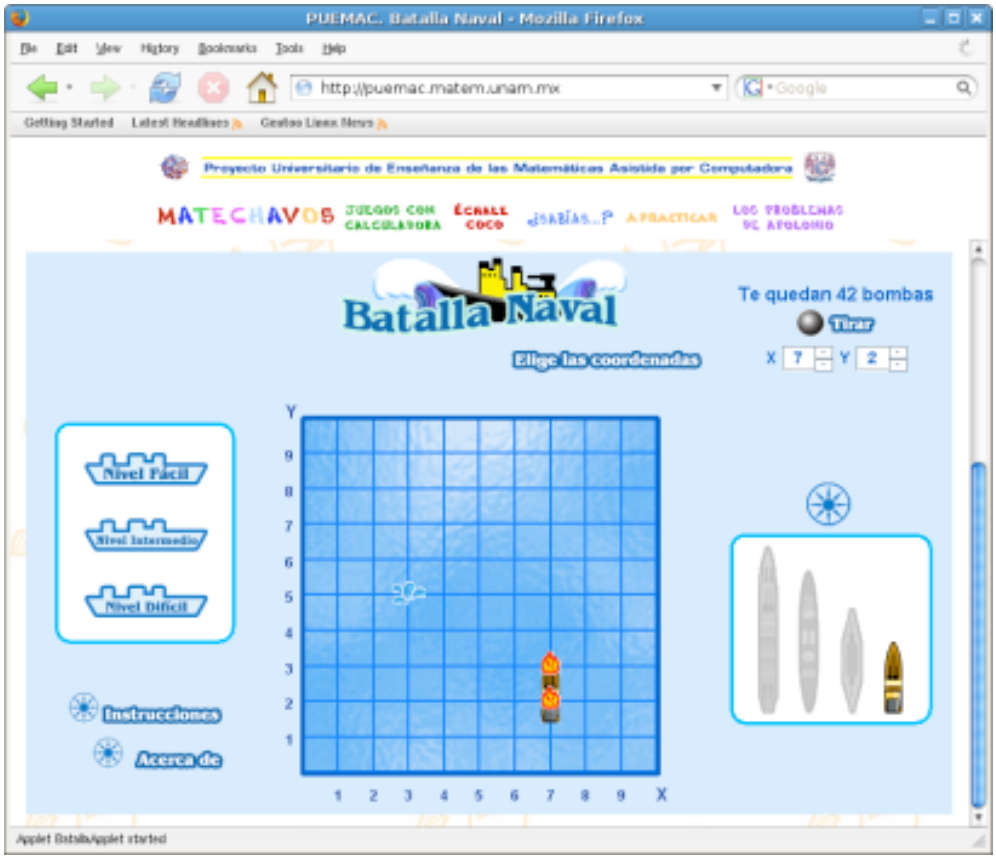


Figura 2.11: Batalla naval

2.5.3 Implementación

La figura 2.12 muestra el diagrama UML de las clases utilizadas en el juego de Batalla Naval. En este juego necesitamos conocer en todo momento la posición de los barcos y dónde se han lanzado bombas, para eso usamos la clase Tablero. Para representar al tablero usamos un arreglo bidimensional; cada entrada del arreglo representa un cuadrado del área de juego y guarda un valor entero. Este valor indicará si en esa casilla hay un barco, de qué tamaño es el barco, si ya se ha tirado una bomba o si está libre. La clase barco tiene como atributos el tamaño, su posición dentro del tablero, su orientación (vertical u horizontal) y un arreglo que indica cuántos bom-

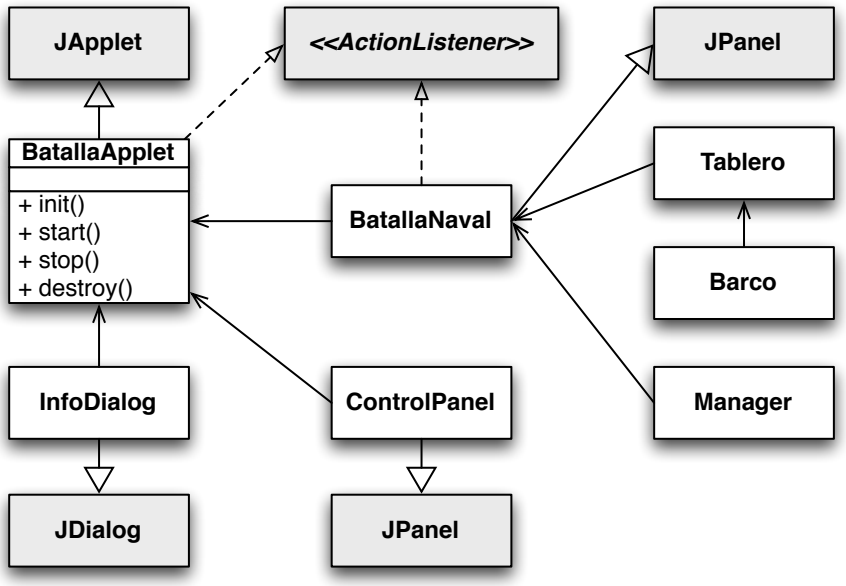


Figura 2.12: Diagrama de clases del *applet* Batalla naval

bazos ha recibido. Cuando el número de impactos es igual a la longitud del barco, el barco cambiara al estado visible para que sea pintado en el tablero y se considera hundido.

La clase Manager se encarga de procesar y leer las coordenadas que seleccionó el usuario, modificar el tablero en esa posición y en caso de haber golpeado a un barco, incrementar el número de impactos del barco correspondiente. Después de eso verifica si ya se hundieron todos los barcos o si se terminaron los tiros para revisar si terminó el juego.

Por último en la clase BatallaNaval se pinta el tablero y los barcos. Como se mencionó en la sección 2.2.3 el algoritmo de animación es el mismo en todas las aplicaciones.

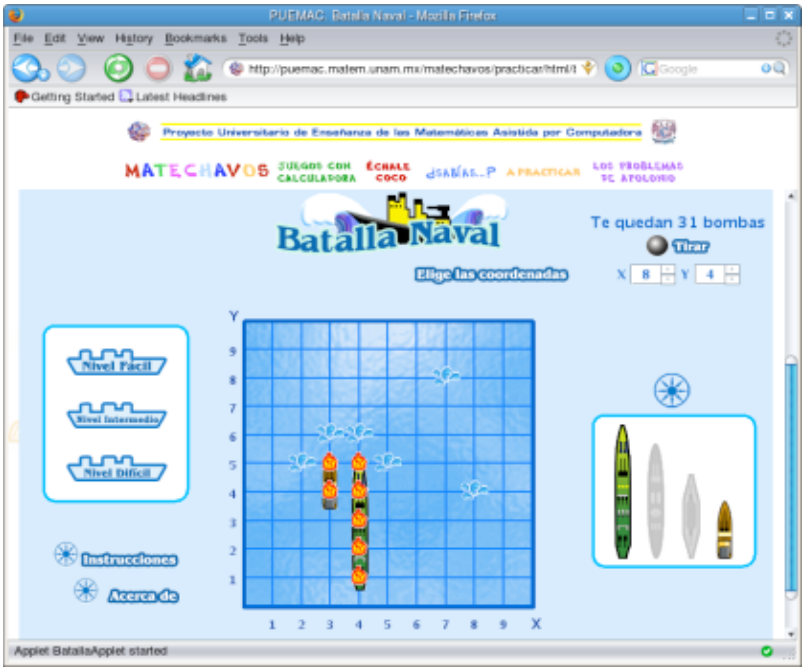


Figura 2.13: Batalla naval

Resultados 3

Cada una de las aplicaciones mostradas en este trabajo fueron llevadas hasta las aulas de clase de algunas escuelas públicas y privadas, con la intención de evaluar que cumplieran con los objetivos iniciales. Como resultado de este ejercicio se identificaron las modificaciones necesarias para mejorarlas. Este punto se logró gracias al apoyo del grupo de pedagogas del departamento de cómputo para niños de la DGSCA.

Por otro lado, las aplicaciones no sólo han sido herramientas de apoyo para aprender y practicar matemáticas en Internet y en las aulas de clase, sino también han servido como base para varios artículos de investigación en las áreas de docencia de las matemáticas y pedagogía. Además, cabe mencionar que el juego de globos representó parte fundamental de una tesis de la licenciatura en Matemáticas presentada en 2005.

A continuación listamos las referencias de los artículos y la tesis mencionada:

- Martínez, P. y Cazares M. A. (2004) *"A romper globos. Un juego interactivo para trabajar fracciones."* En
  - *XXXVII Congreso Nacional de la Sociedad Matemática Mexicana.* Ensenada, Baja California 10-15 de octubre 2004.
  - *XX Simposio Internacional de Computación en la Educación.* Puebla, Puebla, 16 al 20 de octubre 2004.

  - Congreso Internet 2004 DGSCA-UNAM, 4 y 5 de noviembre 2004.

- Martínez, P. y González G. (2005) *"Las fracciones en la recta. Procedimientos de resolución desarrollados por niños sordos". XXI Simposio Internacional de Computación en la Educación.* Hermosillo, Sonora, 1○ al 5 de octubre 2005.

- Cázarez M. (2005) *"La enseñanza de las fracciones en la educación básica. Aplicación y análisis de una situación didáctica en sexto grado de primaria".* Tesis presentada por Cázarez Ávila M. A. en la Facultad de Ciencias de la UNAM en 2005, para obtener el título de Matemática, bajo la dirección y asesoría de la M. en C. Norma Patricia Martínez Falcón y la Mat. Julieta del Carmen Verdugo Díaz, respectivamente.

Otros proyectos en los que se han visto involucradas estas aplicaciones fueron:

- En el año 2006 la Secretaría de Educación Pública (SEP) seleccionó las aplicaciones Globos y Cuadrados Mágicos, para ser incluidas como parte de una herramienta computacional creada para estimular el proceso de enseñanza y aprendizaje, llamada Enciclomedia, la cual ha sido difundida a nivel nacional en todas las escuelas de enseñanza básica, de primero a sexto grado.

- En el año 2007 la cuatro aplicaciones fueron seleccionadas como material de apoyo para diversos reactivos de la Evaluación Nacional del Logro Académico en Centros Escolares (ENLACE), cuyo objetivo es retroalimentar a padres de familia, estudiantes, docentes, directivos y autoridades educativas con información para mejorar la calidad de la educación.

  En sexto grado se utilizaron los siguientes juegos para apoyar algunos de los contenidos que se trabajan.

  Batalla naval: Con este juego se trabaja la ubicación de puntos en el plano (un reactivo).

  En quinto grado se utilizaron los siguientes juegos.

  La pulga y las trampas: Este juego es un apoyo para practicar multiplicaciones de dígitos y multiplos comunes (tres reactivos).

  Cuadrados mágicos: Con este juego se trabaja la suma de números para obtener un resultado específico (un reactivo).

A romper globos: Este juego se utilizó para apoyar reactivos sobre ubicación de fracciones en una recta, el orden de las fracciones, fracciones equivalentes y comparación de fracciones (tres reactivos).

Batalla naval: Con este juego se trabaja la ubicación de puntos en el plano cartesiano (un reactivo).

Para cuarto grado se usó La pulga y las trampas en cuatro reactivos, Cuadrados mágicos en cinco reactivos, A romper globos en tres y Batalla naval en uno.

En tercer grado se usaron La pulga y las trampas y Cuadrados mágicos, en uno y dos reactivos respectivamente.

Todo lo anterior muestra que lo expuesto en este trabajo ha tenido una buena aceptación e impacto dentro del ámbito de la enseñanza de las Matemáticas a nivel nacional.

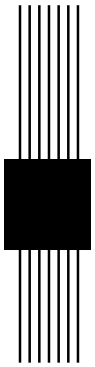
Conclusiones 4

El colaborar en el proyecto PUEMAC fue una gran oportunidad para aplicar los conocimientos de programación y análisis que adquirí durante mis estudios de licenciatura. Sin embargo para cumplir los requerimientos de las aplicaciones también fue necesario aprender nuevos conceptos, por ejemplo las técnicas que se usan en el desarrollo de aplicaciones gráficas y juegos, pues de esto no tenía mucho conocimiento.

El planteamiento y desarrollo de las aplicaciones educativas realizadas requerían de la colaboración de un grupo interdisciplinario (computólogos, pedagogos, matemáticos y diseñadores gráficos), lo cual me permitió interactuar con profesionistas de otras áreas. Esta experiencia fue bastante agradable y fructífera, ya que me hizo considerar diversas perspectivas durante el desarrollo de las aplicaciones.

El trabajo realizado me permitió potenciar aptitudes, habilidades, conocimientos y herramientas de desarrollo que considero indispensables en la formación intelectual y académica de un profesional de las ciencias de la computación, como por ejemplo la manera en que se debe llevar el control de software y el estar preparado a cambios en la especificación inicial del problema.

Bibliografía



Brackeen, D. (2003). *Developing Games in Java*. New Riders.

Brousseau, G. (1994). Los diferentes roles del maestro. In *Didáctica de matemáticas. Aportes y reflexiones*. Paidós educador 112.

Campione, M., Walrath, K., and Huml, A. (2001). *The Java(TM) Tutorial: A Short Course on the Basics*. The Java Series. Prentice Hall, 3rd edition.

Charnay, R. (1994). Aprender por medio de la resolución de problemas. In *Didáctica de matemáticas. Aportes y reflexiones*. Paidós educador 112.

Davison, A. (2005). *Killer Game Programming in Java*. O'Reilly.

Dávila, M. (2002). Las situaciones de reparto para la enseñanza de las fracciones. aportes para la elaboración de un estado del conocimiento. Master's thesis, Centro de investigación y estudios avanzados del IPN.

Elliott, J., Loy, M., and Wood, D. (2002). *Java Swing*. O'Reilly, 2nd edition.

Fan, J., Ries, E., and Tenitchi, C. (1996). *Black Art of Java Game Programming*. Waite Group Press.

Gheorghies, O. (2006). Metauml: Tutorial, reference and test suite.

Gálvez, G. (1994). La didáctica de las matemáticas. In *Didáctica de matemáticas. Aportes y reflexiones*. Paidós educador 112.

Knudsen, J. (1999). *Java 2D Graphics*. O'Reilly.

Marinacci, J. and Adamson, C. (2005). *Swing Hacks: Tips and Tools for Killer GUIs*. O'Reilly.

Niemeyer, P. (2005). *Learning Java*. O'Reilly.

Parra, C. (1994). Cálculo mental en la escuela primaria. In *Didáctica de matemáticas. Aportes y reflexiones*. Paidós educador 112.

Saíz, I. (1994). Dividir con dificultad o la dificultad de dividir. In *Didáctica de matemáticas. Aportes y reflexiones*. Paidós educador 112.

Walrath, K., Campione, M., and Huml, A. (2004). *The JFC Swing Tutorial: A Guide to Constructing GUIs*. The Java Series. Prentice Hall.