



USAC
TRICENTENARIA
Universidad de San Carlos de Guatemala

Universidad De San Carlos De Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Estructura de datos A
Ing. Luis Fernando Espino
Aux. Robinson Jonathan Perez Fuentes



Proyecto
Segundo semestre 2017 vacaciones diciembre

Contenido

Proyecto	3
Objetivos	3
Descripción del Proyecto.....	3
Descripción del Juego.....	3
Tableros.....	3
Unidades	3
Desarrollo del juego	5
Movimientos	6
Implementación	6
Servidor del juego	7
1ra Fase	7
2da. Fase.....	7
Cliente	8
Cliente Android (Opcional).....	8
Estructuras de Datos	8
Usuarios.....	8
Lista de Juegos.....	9
Tablero de Juego	9
Historial de juego	10
Contactos	10
Dispersión de usuarios.	11
Envío de correos.....	11
Tecnologías.....	11
Reportes	11
Graphviz	11
Consultas	12
Graphviz	12
Consultas	12
Entregables.....	12
Observaciones	13

Proyecto

Objetivos

- Adquirir los conocimientos fundamentales en la aplicación e implementación de las estructuras de datos en un ambiente comercial.
- Crear una aplicación utilizable en un ambiente comercial, tomando en cuenta la realización de los entregables y la recopilación de información para el desarrollo de la misma.
- Implementar los conceptos referentes a punteros y memoria dinámica en la solución de problemas.
- Aplicar los conceptos y estructuras vistas en la resolución de problemas reales.

Descripción del Proyecto

En la actualidad el mercado de software, no se restringe únicamente al desarrollo de sistemas administrativos, ya que en el mismo se tiene un gran negocio con el desarrollo de videojuegos, los cuales presentan modelos de monetización que los hacen rentables a través de micro transacciones que permiten avanzar en el juego. El proyecto consiste en desarrollar una versión del juego Advance Wars el cual llevara por nombre **Naval Wars**, utilizando diferentes tecnologías como servicios web y Android.

Descripción del Juego

Naval wars está basado en el videojuego para Game Boy Advance, lanzado en 2001, conocido como Advance Wars, pero con la diferencia que todo se sitúa en un ambiente marítimo y aéreo. El objetivo del juego es destruir todas las unidades del oponente haciendo movimientos en un tablero.

Tableros

Cada jugador maneja a sus unidades en un tablero dividido en casillas. Este tablero puede tener hasta 4 niveles, por lo tanto, será un cubo disperso. Las unidades de cada jugador serán colocadas sobre el tablero en el cual podrán moverse y efectuar un ataque por unidad, si es posible.

Unidades

Al comenzar cada jugador posicionara sus unidades en el tablero, pudiendo colocarlas solamente hasta la mitad del tablero, de forma que se eviten colisiones al intentar colocar una unidad en un espacio que ya este ocupado por el oponente.

Existen 4 niveles para las unidades, los cuales se identifican de la siguiente forma:

- Satélite
- Avión
- Barco

- Submarino

Para cada nivel existen diferentes tipos de unidades, con sus características de movimiento. Alcance, vida etc. En la tabla siguiente se describen estas unidades y su información

Nivel: Satélites					
Unidad	Movimiento	Alcance	Daño	Vida	Observaciones
Neosatelite	6	0	2	10	Puede atacar únicamente a las unidades que se encuentren por debajo de el
Nivel: Aviones					
Unidad	Movimiento	Alcance	Daño	Vida	Observaciones
Bombardero	7	0	5	10	Puede atacar únicamente a las unidades que se encuentren por debajo de el
Caza	9	1	2	20	Puede atacar a cualquier unidad dentro de su rango de alcance
Helicóptero de combate	9	1	3	15	Puede atacar a cualquier unidad dentro de su rango de alcance
Nivel: Barcos					
Unidad	Movimiento	Alcance	Daño	Vida	Observaciones

Fragata	5	2-6	3	10	Puede atacar a cualquier unidad dentro de su rango de alcance
Crucero	6	1	3	15	Puede atacar a cualquier unidad dentro de su rango de alcance
Nivel: Submarinos					
Unidad	Movimiento	Alcance	Daño	Vida	Observaciones
Submarino	5	1	2	10	Esta unidad no será visible para el oponente y podrá atacar cualquier unidad dentro de su rango de alcance

Desarrollo del juego

Al iniciar el juego, se debe definir del lado del servidor, los parámetros del juego los cuales serán:

- Número de unidades por nivel
- Tamaño del tablero
 - Las dimensiones se tomaran únicamente para las coordenadas en X y Y, ya que la profundidad es una constante de valor 4.
- Tipo de juego
 - Normal
 - El juego termina hasta que uno de los dos jugadores destruye todas las unidades de su oponente
 - Tiempo
 - El juego termina en un tiempo definido, se determina ganador al jugador que posea más unidades, en caso de que posean la misma cantidad se declarara como empate.
 - Base
 - Cada jugador tendrá una base que deberá colocar en el tablero, el jugador que capture primero la base del oponente será el ganador, también se

declarara ganador al que destruya primero todas las unidades del oponente.

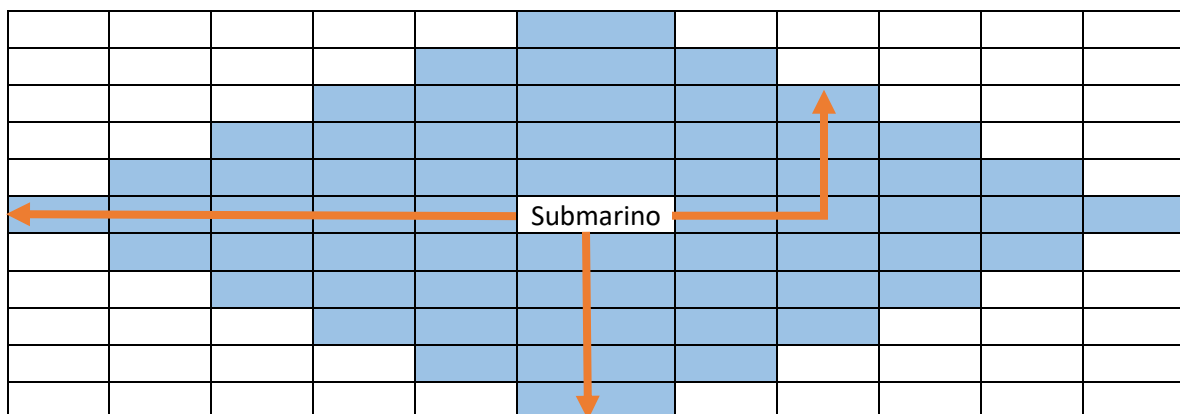
Toda vez se han establecido los parámetros de juego anterior, se dispondrá de que cada jugador coloque sus unidades en el tablero, como ya se mencionó dichas unidades no podrán ser colocadas más allá de la mitad del tablero para evitar colisiones con el oponente, es decir que al inicio del juego a cada jugador le corresponde una mitad del tablero, para colocar sus unidades y su base, si es el caso.

Una vez colocadas las unidades, se dará inicio a una serie de rondas. En cada ronda cada jugador en su turno podrá mover sus unidades, según estas lo permitan, indicando la unidad que se desea mover y la coordenada a la que se desea hacerlo, el jugador durante su turno podrá indicar la unidad con la que desea atacar y la coordenada donde desea hacerla siempre que el alcance de la misma lo permita, se debe tomar en cuenta que las unidades solo se podrán mover y atacar una vez por turno; se da por terminado el turno del jugador cuando este ya ha movido y atacado con todas sus unidades o cuando él decida finalizar su turno, dando paso a que comience el turno del oponente.

En la interfaz se deben mostrar las unidades, junto con su información de vida que cada una aún posee, adicional se deberá agregar una bitácora que muestre las incidencias del juego. El diseño de la misma queda a discreción del estudiante, pero debe mostrar todas y cada una de las opciones indicadas.

Movimientos

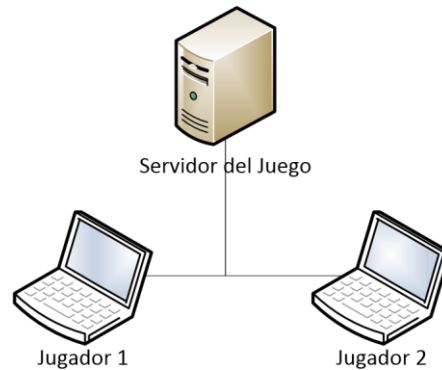
Los movimientos se podrán realizar en forma horizontal y vertical o una combinación de ambos, según le sea posible a la unidad, de forma que no podrá moverse en diagonales, guiarse por el ejemplo adjunto del rango de movimiento de la siguiente unidad.



Implementación

El juego se tendrá en un servidor que estará a disposición a través de Web Services (.Net), por lo que será necesario un programa cliente. Este programa será una aplicación web

(.Net). Por lo tanto, es necesario que la comunicación se realice a través de red, de la siguiente forma:



Para desarrollar este diagrama de red, se utilizarán máquinas virtuales. El orden y configuración de las mismas queda a criterio del estudiante.

Servidor del juego

Este equipo tendrá el servidor del juego, así como las estructuras necesarias para el desarrollo del juego.

Se debe proveer una página de administración. Debe tener una pantalla de login para validar el ingreso de un usuario administrador. Solamente este usuario puede acceder a esta parte del sistema.

Dentro de la funcionalidad, debe tener lo siguiente:

1ra Fase

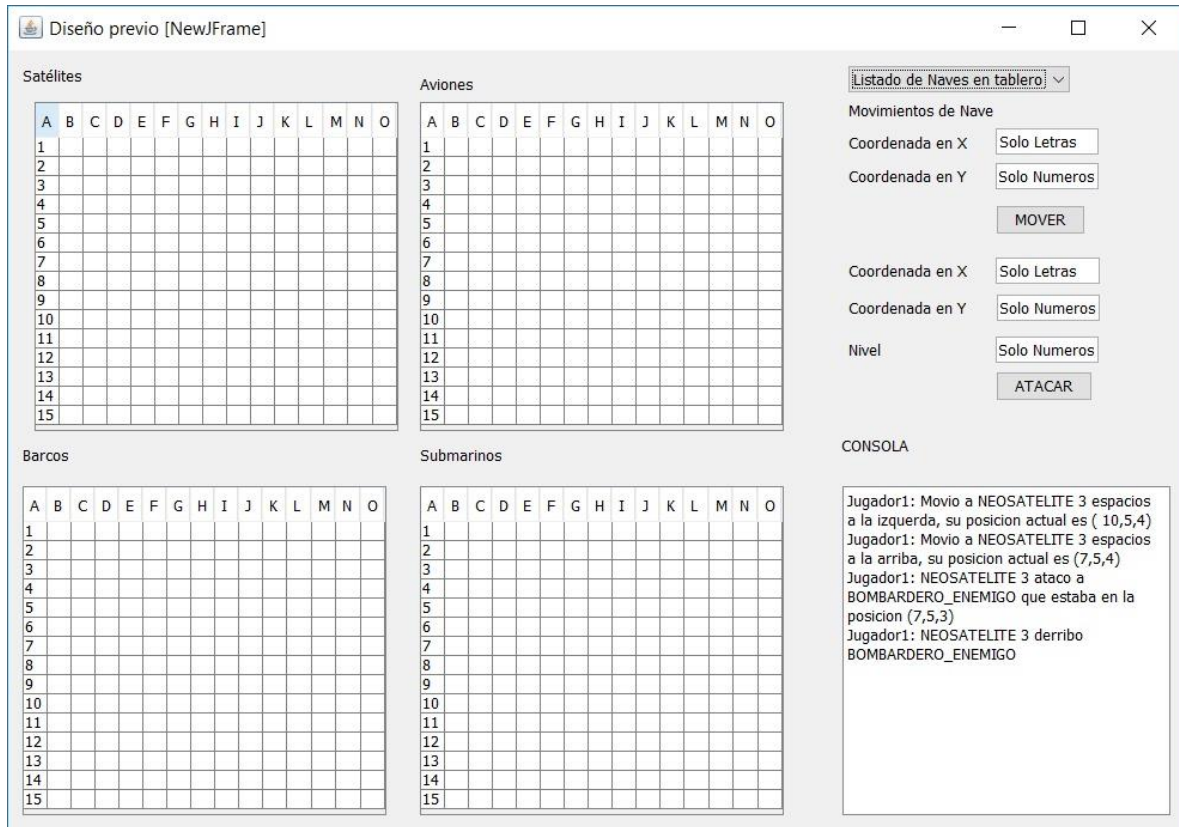
- ABC de usuarios
- ABC juego
 - Se dan dos usuarios y se crean las estadísticas de juego.
- Carga de datos
- Ver reportes

2da. Fase

- ABC de contactos
- ABC de movimientos

Cliente

Tiene una aplicación de tipo cliente del juego. El siguiente es un prototipo de la interfaz.



Se debe tener una consola que informe el turno activo, los movimientos de las unidades, las unidades destruidas propias y del oponente, así como el progreso del tiempo, si ese tipo de juego es el escogido.

Debe realizarse un login en cada cliente, tomando como base los usuarios en el árbol.

Cliente Android (Opcional)

Este cliente es exclusivo de la segunda fase. Este cliente tendrá, de igual forma, un login tomando en cuenta el árbol de usuarios. La funcionalidad será la siguiente:

- Enviar correos
- Ver cola de contactos a enviar correos
- Ver pila de correos recibidos
- Ver todos reportes respectivos de la segunda fase

Estructuras de Datos

Usuarios

Los usuarios se manejarán con un **árbol binario de búsqueda** por nickname. Del usuario se necesita guardar:

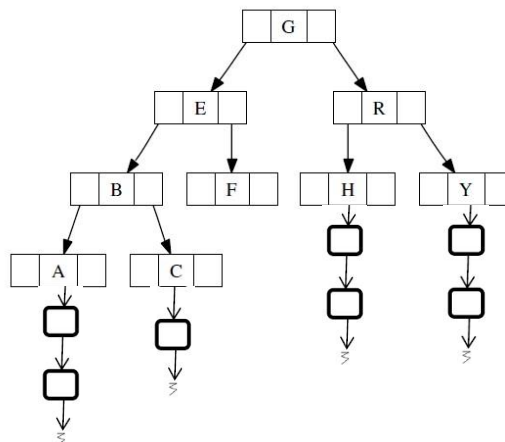
- Nickname (el cual debe ser único)
- Contraseña
- Correo electrónico
- Variable que indica si está conectado o no
- Apuntador a la lista de juegos

Lista de Juegos

Es una **lista doblemente enlazada** que contiene los siguientes datos del juego:

- Nickname del oponente
- Unidades desplegadas
- Unidades sobrevivientes
- Unidades destruidas
- Una bandera que indica si ganó o perdió la partida

La estructura se verá de la siguiente forma:



Tablero de Juego

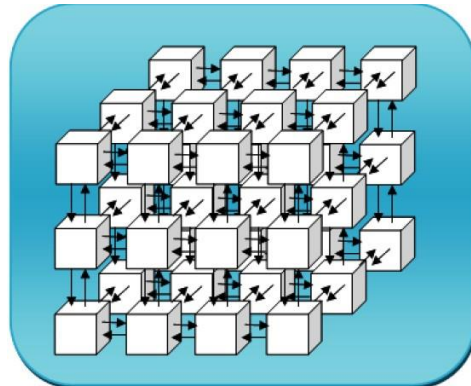
El tablero tendrá la forma de un cubo disperso y se manejará en una sola matriz para ambos jugadores, tomar en cuenta que, en el caso de los submarinos, la imagen que se muestra al propietario del submarino, el mismo debe ser visible, y la que se le muestra al oponente no debe ser visible dicha unidad.

La malla principal de la matriz (la que tiene los encabezados) es la malla que maneja los barcos. De allí se apuntará a las siguientes dimensiones:

Satélites
Aviones
Barcos
Submarinos

En cada nodo se mostrará la unidad que se encuentra allí con un identificador único, junto con la información de vida que tiene, el movimiento, alcance y daño que puede realizar.

Para el mejor manejo de la información, se sugiere realizarlo de la siguiente forma:



Historial de juego

Este historial es exclusivo de la segunda fase del proyecto. El historial consiste en guardar cada uno de los ataques realizados en el juego, en un árbol B. Los datos por guardar son los siguientes.

- ▲ Coordenada X
- ▲ Coordenada Y
- ▲ Unidad atacante
- ▲ Resultado (Daño, eliminación del objetivo)
- ▲ Tipo de unidad dañada
- ▲ Emisor
- ▲ Receptor
- ▲ Fecha
- ▲ Tiempo restante (si fuera el caso)
- ▲ Numero de ataque

Antes de iniciar el juego, se debe preguntar el parámetro por el que se debe ordenar el árbol B. Se debe tomar en cuenta que el mismo puede repetirse. Si se diera el caso, el segundo criterio de ordenamiento es el número de tiro.

Contactos

Los contactos son exclusivos de la segunda fase del proyecto. El usuario, además de tener su lista de juegos, se agregará un árbol AVL de contactos. Estos contactos pueden ser usuarios ya existentes en el sistema o usuarios que no existan. Si el usuario ya existe, el nodo AVL solo apuntará al nodo del usuario ya existente, en caso contrario, se debe ingresar todos los datos en un nuevo nodo AVL. Los datos son los mismos del usuario. Entonces, el contenido del nodo AVL sería el siguiente:

- Apuntador a usuario (si ya existe)
- Nickname
- Contraseña

- Correo electrónico

Dispersión de usuarios.

La dispersión de usuarios es exclusiva de la segunda fase del proyecto. Teniendo los usuarios en el árbol binario, se tomará todos y cada uno de ellos y se exportaran a una tabla de dispersión hash. Esta tabla hash tendrá los siguientes parámetros:

- Tamaño inicial: 43
- Porcentaje de ocupación máximo: 50%
- Porcentaje de ocupación mínimo: 30%
- Estrategia de resolución de colisiones: Exploración cuadrática
- Función de dispersión: Plegamiento y modular por el tamaño de la tabla

Envío de correos

El envío de correos es exclusivo de la segunda fase del proyecto. Tomando en cuenta el árbol de contactos, se podrá enviar por correo las estadísticas del juego. Estas estadísticas son las mismas que se guardan en la lista de juegos.

Para enviar los correos, se debe colocar en una **cola** los usuarios a los que se enviará en correo. Luego de enviarlo, el usuario debe cargar sus correos a la interfaz a través de una **pila**. Solamente deben tomarse los correos no leídos. Estos se leerán a través de la pila, vaciando la estructura.

Tecnologías

La tecnología será dividida de la siguiente forma:

Primera Fase

- Servidor (asp)
 - Disponibilidad a través de web services
- Cliente (asp)

Segunda Fase

- Cliente (Android) (Opcional)
 - A través de este nuevo cliente se podrá ver la funcionalidad de la segunda fase, consumiendo los servicios web provistos por el servidor.
- Servidor (ASP)

Reportes

1ra fase

Servidor

Graphviz

- Tablero de juego: Muestra el tablero en el estado actual.

- Tablero inicial: Muestra cómo estaba el tablero al ser iniciado el juego.
- Unidades sobrevivientes: Muestra el tablero solamente con las unidades que sobrevivieron.
- Unidades destruidas: Muestra el tablero solamente con las unidades que fueron destruidas.
- Usuarios, estructura completa con sus listas.
- Árbol espejo de usuarios
- Mostrar los datos del árbol
 - o Altura
 - o Niveles
 - o Cantidad de nodos hoja
 - o Cantidad de nodos rama

Consultas

- Top 10 de usuarios con más juegos ganados
- Top 10 de usuarios con mayor porcentaje de unidades destruidas

2da. Fase

Servidor

Graphviz

- Historial de tiros del juego (Árbol B)
- Árbol de contactos de algún usuario requerido
- Usuarios (Hash)
- Cola de contactos a enviar correo
- Pila de correos recibidos

Consultas

- Top 10 de usuario con más unidades eliminadas
- Top 10 de usuarios con más contactos
- Mostrar juego con más ataques realizados para su finalización
- Mostrar juego con menos ataques realizados para su finalización

Entregables

- Código Fuente
- Ejecutable
- Instalador
- Manual Técnico

Observaciones

- ▲ Lenguaje de programación a utilizar: C#
- ▲ Sistema Operativo: Libre.
- ▲ IDE: Libre.
- ▲ Herramientas de reportes: Graphviz
- ▲ Tecnologías: ASP, Web Services y Android
- ▲ Las graficas deben ser abiertas por la aplicación, no buscarlas en el sistema. Se penalizará con 10 puntos si no se cumple este requerimiento.
- ▲ La aplicación que no genera reportes gráficos tendrá nota de cero puntos.
- ▲ No se permite el uso de librerías que manejen las estructuras de datos, estas deben ser creadas por el estudiante. Si se utilizan estas librerías, el proyecto tendrá una nota de cero puntos.
- ▲ Proyectos con excepción NULL tendrán nota de cero puntos.
- ▲ Se debe realizar utilizando la arquitectura mostrada. De no ser así, se tendrá una penalización de 30 puntos.
- ▲ No será permitido el uso de ningún tipo de base datos, para guardar la información, toda será almacenada en las estructuras respectivas.
- ▲ Copias tendrán nota de cero puntos y serán reportadas al catedrático y a la escuela de sistemas.
- ▲ Fecha y modo de entrega:
 - ❖ Fase 1
 - Miércoles 27 de diciembre
 - ❖ Fase 2
 - Lunes 8 de enero
 - ❖ La entrega se realizará vía GitHub antes de las 11:59 p.m. para cada fase deberán crear una carpeta en su repositorio y cargar su proyecto, la carpeta llevara por nombre: proyecto_fase1 y Proyecto_fase2, según corresponda.