

CB100 - Algoritmos y Estructuras de datos

Cátedra Ing. Schmidt Gustavo - 1er cuatrimestre 2025

Trabajo Práctico 2: INVASIÓN GALÁCTICA V1.0

Objetivo

Desarrollar un juego de estrategia por turnos en el que los jugadores compiten por el control de un sector del espacio, desplegando flotas, atacando bases enemigas y usando tecnologías avanzadas.

Enunciado

Invasión Galáctica es un juego de **N jugadores**, donde cada uno comienza con una base espacial y una cantidad de naves en un tablero tridimensional de **$X \times Y \times Z$** sectores espaciales. El objetivo es eliminar las bases de los oponentes y tomar el control del sector.

En cada turno, un jugador puede:

- a) **Robar una carta** del mazo de tecnología y decidir si usarla o guardarla.
- b) **Atacar un sector** con una nave de combate. Si hay una base enemiga, se reduce su escudo. Si el escudo llega a cero, la base es destruida.
- c) **Colocar un satélite espía** en un sector. Si un jugador mueve una nave a ese sector, el propietario del satélite recibe información sobre el movimiento. Si dos satélites coinciden, ambos son destruidos.
- d) **Mover una nave** a otra posición del tablero. Si se mueve a un sector con una base enemiga debilitada, la puede conquistar. No se pueden mover naves a sectores con radiación activa por explosiones recientes.
- e) **Generar alianzas**: definir un modo de hacer alianzas e intercambiar beneficios con ella.

Cartas disponibles

El juego tiene **8 cartas**, 4 predefinidas y 4 que los jugadores diseñarán. Las cartas dadas son:

1. **Campo de fuerza:** Aumenta la defensa de una base durante un número de turnos.
2. **Rastreador cuántico:** Revela si hay naves enemigas en un radio de L (valor random) sectores desde la posición elegida.
3. **Doble salto hiperespacial:** Permite mover una nave dos veces en el mismo turno.
4. **Base adicional:** permite agregar una base adicional.

Condiciones de victoria

El juego finaliza cuando solo un jugador conserva su base espacial. Ese jugador es el ganador.

Además se debe poder guardar el estado de una partida para retomarlo a futuro en una nueva instancia del juego.

Interfaz de usuario

Toda la interfaz de usuario debe estar basada en texto. El estado del tablero tiene que mostrarse usando un archivo de imagen BMP. Después de cada turno, el programa debe exportar el tablero en un archivo bitmap con el estado del tablero, de manera que quede una secuencia de imágenes.

No es necesario que se limpie la pantalla, simplemente escribir el estado del tablero luego de cada jugada.

Cuestionario

Responder el siguiente Cuestionario:

- 1) ¿Qué es un svn?
- 2) ¿Que es una "Ruta absoluta" o una "Ruta relativa"?

3) ¿Qué es git?

Normas de entrega

Trabajo práctico grupal: 6 personas. Cada grupo deberá definir un nombre.

Reglas generales: respetar el Apéndice A.

Se deberá subir un único archivo comprimido al campus por grupo, en un link que se habilitará para esta entrega. Este archivo deberá tener un nombre formado de la siguiente manera:

NombreDelGrupo-TP2.zip

Deberá contener los archivos fuentes (no los binarios), el informe del trabajo realizado, las respuestas al cuestionario, el manual del usuario y el manual del programador (Todo en el mismo PDF). El nombre del grupo lo define el grupo.

La fecha de entrega vence el día jueves 13/06/24 a las 23.59hs.

Se evaluará: funcionalidad, eficiencia, algoritmos utilizados, buenas prácticas de programación, modularización, documentación, gestión de memoria y estructuras de datos.

Apéndice A

1) Usar las siguientes convenciones para nombrar identificadores.

- a) Clases: Los nombres de clases siempre deben comenzar con la primera letra en mayúscula en cada palabra, deben ser simples y descriptivos. Se concatenan todas las palabras. Ejemplo: Coche, Vehiculo, CentralTelefonica.
- b) Métodos: Deben comenzar con letra minúscula, y si está compuesta por 2 o más palabras, la primera letra de la segunda palabra debe comenzar con mayúscula. De preferencia que sean verbos. Ejemplo: arrancarCoche(), sumar().
- c) Variables y objetos: las variables siguen la misma convención que los métodos. Por Ejemplo: alumno, padronElectoral.

- d) Constantes: Las variables constantes o finales, las cuales no cambian su valor durante todo el programa se deben escribir en mayúsculas, concatenadas por "_". Ejemplo: ANCHO, VACIO, COLOR_BASE.
- 2) Si el trabajo práctico requiere archivos para procesar, entregar los archivos de prueba en la entrega del TP. Utilizar siempre rutas relativas y no absolutas.
 - 3) Entregar el informe explicando el TP realizado, manual de usuario y manual del programador.
 - 4) Comentar el código. Todos los tipos, métodos y funciones deberían tener sus comentarios.
 - 5) Modularizar el código. No entregar 1 o 2 archivos, separar cada clase.
 - 6) No utilizar variables globales.
 - 7) El tablero no puede ser `Sector[][][] tablero = new Sector[10][10][10];`
 - 8) Si cualquier estructura de control tiene 1 línea, utilizar {} siempre, por ejemplo:

```
for(int i = 0; i < 10; i++) {  
    std::cout << i;  
}
```