

# ESCUELA COLOMBIANA DE INGENIERIA

## PROGRAMACIÓN ORIENTADA A OBJETOS

### Scorched ECI 2009-02

[Colaboración de Ferney Velasquez y Javier Quintero]

#### DESCRIPCIÓN

---

"**Scorched Earth** fue un popular juego de computadora en la era de DOS, en el cual se jugaba a controlar tanques en un sistema de juego por turnos, sobre un terreno en dos dimensiones. Cada jugador debía ajustar el ángulo y la potencia de disparo del tanque bajo su control, lo cual se realiza todos los turnos." <http://www.youtube.com/watch?v=Io3LjftP6YQ>

El objetivo del proyecto **Scorched ECI**, inspirado en **Schorched Earth**, es desarrollar una aplicación para la batalla entre dos tanques, uno dirigido por el usuario y otro por el computador. En **Scorched ECI** los tanques realizan sus movimientos por turnos; la batalla termina cuando un tanque destruye a su oponente, o se rinde para evitar su destrucción.

#### Campo de batalla

El campo de batalla es un espacio rectangular de celdas que puede ser un obstáculo, estar vacíos o contener un tanque. Los obstáculos más importantes de **Scorched ECI** son las montañas formadas por celdas de tierra. Cuando una celda de tierra es desintegrada, todo lo que estaba encima de ella cae.

#### Tanques

Cada tanque está equipado con diferentes tipos de armas, un tanque de combustible y un sensor de daño. Las armas y el tanque de combustible también se carga al iniciar la batalla. El sensor de daño indica si el nivel actual de destrucción del tanque.

El tanque dirigido por el computador puede ser de uno de los siguientes perfiles:

- ✓ Azaroso: decide sus actuaciones al azar
- ✓ Destructor: está programado para destruir a su oponente, sin preocuparse por sí mismo.
- ✓ Protector: su principal interés es protegerse; normalmente, busca esconderse en los obstáculos del campo de batalla.

#### Armas

Los tanques pueden tener diferentes tipos de armas.

- ✓ Proyectiles : tienen un recorrido en línea recta en el ángulo de disparo, para cuando golpea contra algo.
- ✓ Bombas : se mueven en línea recta tres unidades en el ángulo de disparo y después caen perpendicularmente.
- ✓ Misiles : se alejan cinco unidades en el ángulo de disparo buscando al objetivo; si lo encuentra en esta área de cubrimiento se dirige directamente hacia él.

#### Movimientos

Las acciones básicas de un tanque son:

- ✓ Avanzar: El avance solo se puede realizar si tiene la cantidad suficiente de combustible para trasladarse el número de posiciones que se quiere, el tanque puede trepar pero este movimiento requiere gran cantidad de gasolina. Para avanzar, se indica el número de posiciones a recorrer, en una dirección indicada (este, oeste) y no se permite que un jugador se mueva a la celda previamente ocupada.
- ✓ Atacar: Envía un arma en una dirección determinada (0, 180).

En cada movimiento cada tanque puede gastar un número máximo de combustible y de explosivos.

**Impacto**

Cuando un tanque o una celda de tierra recibe el impacto de una bala el daño que sufre depende del tipo de arma (proyectil, bomba, misil, nuke, etc) y de la distancia a su agresor. En el caso del tanque, aumenta su nivel de daño; y en el caso de la celda de tierra, puede o no ser desintegrada.

## REQUISITOS FUNCIONALES

---

La aplicación debe:

- Permitir generar aleatoriamente un campo de batalla dado ancho, largo y porcentaje de montañas.
- Permitir leer un campo de batalla dado el nombre de un archivo (sólo para la competencia)
- Permitir identificar al jugador oponente con su nombre (el nombre del equipo).
- Permitir seleccionar el tipo de perfil del tanque computador.
- Permitir indicar cual tanque va a realizar el primer movimiento.
- Presentar permanentemente el estado del campo de batalla.
- Presentar permanentemente la información completa del estado de cada uno de los tanques.
- Permitir que el usuario realice sus movimientos indicando posibles errores. Si el usuario se equivoca pierde el turno.
- Presentar el movimiento del tanque computador de manera "inmediata" (Máximo 5 segundos)
- Decidir cuando termina la batalla y comunicar la causa.
- Permitir terminar la batalla en cualquier momento.

## REQUISITOS DE DISEÑO

---

### De extensión

Permitir generar nuevas versiones de la aplicación que incluyan:

- Nuevos obstáculos (por ejemplo, en un futuro, celdas con árboles los cuales desaparecen tras recibir cualquier impacto)
- Nuevas armas (por ejemplo, en la competencia, las bombas)
- Nuevos perfiles de tanque computador (por ejemplo, en la competencia, el nuevo perfil con la estrategia)

### De visualización

- Los tanques, las armas, los obstáculos y las celdas deben ofrecer una representación texto detallada que permita explorar el estado detallado de cada uno de ellos [toString] .
- Las celdas deben implementar una interfaz **CVisible** con dos métodos: uno para la representación texto y otro para la representación gráfica de ellas en el campo de juego.

### De manejo de errores

- Deben definir mínimo una nueva clase excepción para manejar las excepciones propias.
- Los métodos correspondientes a las acciones deben lanzar una excepción si la acción solicitada no es válida explicando claramente su causa.
- Cuando ocurra una excepción no esperada o una propia grave se debe escribir esta información en el log de errores para los programadores y terminar la ejecución del mismo.

## REQUISITOS DE DOCUMENTACIÓN

---

- Las fuentes deben estar documentadas siguiendo el estándar java (objetivos, parametros, retorno y excepciones)
- Para la primera entrega se deben presentar el diagrama de clases<sup>1</sup> de la capa de aplicación y los diagramas de colaboración<sup>2</sup> de tres métodos : 1. un tanque avanza , 2. un tanque ataca con un arma, 3. un tanque azaroso hace su movimiento
- Para la segunda entrega se deben presentar el diagrama de clases de la capa de aplicación, el diagrama de clases de la capa de presentación y los diagramas de colaboración correspondientes al método: 1. el tanque destructor
- Para la competencia se debe presentar el diagrama de colaboración del nuevo tanque computador realizando su movimiento

---

<sup>1</sup> Para todos los diagramas de clase, únicamente métodos públicos con el encabezado completo.

<sup>2</sup> Para todos los diagramas de colaboración, desde la capa de aplicación con 3 niveles de profundización.

## REQUISITOS DE ENTREGA

---

### Primera entrega [26 de noviembre]

- **Interfaz:** texto
- **Perfiles computador:** azaroso
- **Armas :** misil
- **Campo de batalla:** generación aleatoria.
- **Parámetros: ScorchedECIp**

### Segunda entrega [10 de diciembre]

- **Interfaz:** texto y gráfica
- **Perfiles computador:** azaroso, destructor
- **Campo de batalla:** generación aleatoria.
- **Armas :** misil y bomba
- **Parámetros: ScorchedECIs**

### Competencia [14 de diciembre]

- **Interfaz:** texto y gráfica
- **Perfiles computador:** azaroso, destructor y ?nombre del nuevo perfil
- **Campo de batalla:** generación aleatoria y lectura de archivo
- **Armas :** misil y bomba
- **Parámetros: ScorchedECIc**