Programación I - Segundo semestre de 2015
Trabajo Práctico: Guerra de Hormigas

Las hormigas

Los formícidos, conocidos comúnmente como hormigas, son una familia de insectos eusociales (nivel más alto de organización social que se da en ciertos animales) que, como las avispas y las abejas, pertenecen al orden de los himenópteros. Son uno de los grupos zoológicos de mayor éxito, con cerca de 13000 especies descritos, aunque se estima que pueden ser más de 22000. Se identifican fácilmente por sus antenas en ángulo y su estructura en tres secciones con una estrecha cintura.

Forman grupos de un tamaño que se extiende desde unas docenas de individuos, a millones de individuos. Estos grupos son descritas como superorganismos, dado que las hormigas parecen actuar como una entidad única, trabajando colectivamente en apoyo de la colonia.

Han colonizado casi todas las zonas terrestres del planeta; los únicos lugares que carecen de hormigas indígenas son la Antártida y algunas islas remotas o inhóspitas. Las hormigas prosperan en la mayor parte de estos ecosistemas y se calcula que pueden formar el 15-25 % de la biomasa de los animales terrestres. Se estima que hay entre mil billones ($10^(15)$) y diez mil billones ($10^(16)$) de hormigas viviendo sobre la Tierra. Se considera que su éxito en tantos entornos se debe a su organización social y a su capacidad para modificar hábitats, a su aprovechamiento de los recursos y a su capacidad de defensa.

Colonias y super colonias

Lo más habitual es que las hormigas se muestren agresivas con los miembros de otras colonias y formen colonias simples. Sin embargo, en algunas especies las obreras se mezclan con las de otros hormigueros. Un grupo de colonias donde las hormigas no se exponen a la agresión mutua es conocido como una supercolonia. Las poblaciones de las supercolonias no necesariamente se encuentran en un área contigua.

Hasta el año 2000, la supercolonia de hormigas más grande conocida estaba en la costa de Ishikari en la isla de Hokkaido, Japón. Se estimó que la colonia contenía 306 millones de hormigas obreras y un millón de hormigas reinas viviendo en 45.000 hormigueros interconectados por pasos subterráneos sobre un área de 2,7 km^2.

Comportamiento extraño

Expertos científicos de la biología terrestre de la Universidad Nacional de General Sarmiento han detectado una variación en la conducta de algunas supercolonias. En particular dos, identificadas como la supercolonia Delta y la supercolonia Omega. Estas supercolonias han comenzado a aplicar la técnica del Hormigón Armado en la construcción de sus hormigueros. De este modo, estos cuentan con una estructura mucho más sólida y resistente que antes. Estas supercolonias están disputándose el dominio en una feroz guerra que ocurre a pequeña gran escala y silenciosamente todas las noches en el campus de la universidad entre multitudes de hormigas.

El simulador: Guerra de Hormigas

Para entender mejor el accionar de estos insectos, nos han encargado la construcción de un simulador de guerra entre las supercolonias Delta y Omega. En su primer etapa de desarrollo, el simulador mostrará el terreno de batalla desde un punto de vista aéreo, mostrando la ubicación de las diferentes colonias (hormigueros) pertenecientes a Delta, a Omega o neutrales. Durante un enfrentamiento, cada colonia enviará tropas de hormigas de un lugar a otro con el objetivo de tomar control de todos los hormigueros. Cuando esto suceda, la supercolonia ganadora se coronará victoriosa. Los hormigueros neutrales pueden estar ocupados por hormigas coloradas, las cuales no juegan un rol activo en la guerra pero defenderán su hormiguero de manera férrea.

A continuación se detallan las reglas de la simulación separadas por aspectos:

Las tropas

Las tropas de hormigas consisten de un determinado número de hormigas que avanza en línea recta desde un hormiguero de origen a un hormiguero de destino sin interrupción. Solamente pueden ser perturbados por los insecticidas comentados en la siguiente sección.

Envío de tropas

Un hormiguero puede enviar tropas en cualquier momento, estás órdenes serán dadas por un jugador humano. Las tropas se organizan en el momento de ser enviadas y se conforman por un tercio, dos tercios o el total de la población de un hormiguero (este caso es conocido como el “éxodo”).

Arribo de tropas

Si una tropa arriba a un hormiguero de la misma supercolonia, todas las hormigas de la tropa son bienvenidas y pasan a formar parte del equipo. En cambio, si el hormiguero de llegada no está bajo el control de la misma supercolonia, se libra un combate feroz en el que mueren tantas hormigas de la tropa como hormigas del hormiguero. Es decir, si la tropa tiene mayor cantidad de individuos que la población actual del hormiguero, el hormiguero es conquistado y de la tropa mueren tantos como habitantes tenía el hormiguero al momento de la llegada. Si la tropa tiene menor cantidad, todas las hormigas de la tropa mueren y el hormiguero no cambia de dueño, pero sí decrementa su población tanto como el tamaño de la tropa atacante.

En cualquier caso, la tropa deja de existir como tal, ya que sus integrantes mueren o bien se incorporan a la población del hormiguero.

Si un hormiguero es conquistado por una supercolonia, su nivel se decrementa en 1. Excepto que ya sea de nivel 1.

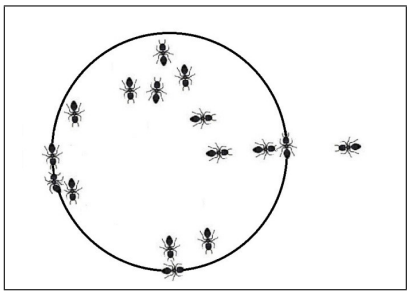
Crecimiento poblacional y estructura de un hormiguero

Cada hormiguero cuenta con una cantidad de hormigas que crece a un determinado ritmo y decrece por la eventual muerte en combate de las hormigas. Cada hormiguero es capaz de enviar tropas a otro hormiguero en cualquier momento.

Los hormigueros de hormigón armado permiten un crecimiento poblacional más grande así como también mejores posibilidades de defensa. Un hormiguero puede estar en uno de los siguientes 4 niveles de construcción:

1. Nivel 1: produce 1 hormiga cada 500 ticks (unidad de tiempo del simulador).
2. Nivel 2: produce 2 hormigas cada 500 ticks.
3. Nivel 3: produce 4 hormigas cada 500 ticks.
4. Nivel 4: produce 8 hormigas cada 500 ticks.

Un hormiguero aumenta de nivel 1 a nivel 2 al alcanzar la población de 40 hormigas, de nivel 2 a nivel 3 al alcanzar las 100, de nivel 3 a nivel 4 al alcanzar las 300.



Insecticidas

Eventualmente, aparecen en el terreno manchas de insecticida. Estas zonas afectadas por productos químicos provocarán que las tropas que las estén atravesando avancen más lentamente que lo normal. La caída de las manchas de insecticida ocurre de manera aleatoria por un período de tiempo aleatorio (entre 1000 y 30000 ticks). De ocurrir una caída de insecticida sobre un hormiguero, el hormiguero pasa a estar infectado. Durante la infección se suspende la creación de nuevas hormigas. Además, la infección es contagiosa. Un hormiguero infectado automáticamente infecta a todos los hormigueros de la supercolonia que tengan nivel exactamente uno menor a él, esto puede generar una infección en cadena que infecte a todos o

gran parte de los hormigueros de la supercolonia. La infección termina cuando desaparece la mancha de insecticida que la provoca.

El terreno

El simulador debe comenzar con una cantidad aleatoria de hormigueros (entre 3 y 13) en la pantalla, dichos hormigueros deberan aparecer de manera aleatoria y dos de ellos pertenecerán de antemano a las supercolonias. Los demás serán hormigueros neutrales, hasta el momento en que sean conquistados. Cabe destacar que los hormigueros neutrales tienen un nivel inicial aleatorio, pero su población de hormigas no aumenta.

Manejo desde el teclado

Durante el transcurso de la guerra, el jugador que comande a la supercolonia Delta tendrá siempre seleccionados a un hormiguero origen y un hormiguero destino. Podrá cambiar su selección utilizando las teclas (A y D) para cambiar la selección de origen, y las teclas W y S para la selección de destino.

Al presionar las teclas 1, 2 y 3, producirá el envío de una tropa desde el hormiguero origen seleccionado al hormiguero destino de tamaño 1, 2 o 3 tercios de la población según indique la tecla presionada.

Análogamente el jugador que comande a la supercolonia Omega podrá seleccionar los hormigueros con las flechas del teclado y enviar tropas con las teclas 8, 9 y 0. Siendo 8 el envío de 1 tercio y 0 el éxodo total.

Es de suma importancia respetar esta configuración del teclado.

Requerimientos obligatorios

Es de carácter obligatorio cumplir con los siguientes requerimientos:

1. El simulador debe permitir realizar el enfrentamiento entre dos supercolonias en un terreno generado inicialmente con 8 hormigueros, siendo solo un subconjunto de ellos controlado por cada supercolonia, el resto deberán estar bajo control neutral.
2. El simulador debe respetar las consignas detalladas anteriormente, en caso de cambios propuestos a las reglas o a la funcionalidad deberán ser consultados previamente con los docentes.
3. Deberá ser posible usar el simulador de a dos jugadores humanos utilizando las teclas del teclado. Recomendamos fuertemente resolver todas las consignas primero para un solo jugador enfrentándose a hormigueros neutrales, y luego cuando todo funcione agregar la funcionalidad del jugador 2.
4. El trabajo se deberá entregar junto con un informe impreso (más información a continuación).

La implementación a entregar debe cumplir como mínimo con todos los requerimientos obligatorios planteados arriba, pero si el grupo lo desea, puede implementar nuevos elementos para

enriquecer la aplicación. Consultar previamente con los docentes por cambios sustanciales.

Requerimientos opcionales

Como requerimiento opcional principal, se quiere que al salir una tropa de hormigas, cada hormiga de la tropa avance a una velocidad autónoma, posiblemente más rápido o más lento que la media. Por ende, que el arribo de la tropa a un hormiguero se realice individualmente al arribar cada hormiga. Las reglas que se aplican serían equivalentes a las de las tropas como si fueran tropas de tamaño 1.

Otro requerimiento opcional es que se simule que las tropas que se defienden tengan más altas chances de ganar el combate que las que están atacando, y además, que haya un leve componente aleatorio en la definición de las batallas.

Para más ideas de funcionalidades extra, consultar con los docentes previamente.

Implementación base

Junto con este enunciado, se les entrega el paquete `entorno.jar` que contiene la clase `Entorno`. Esta clase permite crear un objeto capaz de encargarse de la interfaz gráfica y de la interacción con el usuario. Así, el grupo sólo tendrá que encargarse de la implementación de la "inteligencia" del juego. Para ello, se deberá crear una clase llamada `Juego` que respete la siguiente signatura[1]:

```
public class Juego extends InterfaceJuego
{
    private Entorno entorno;

    // Variables y métodos propios de cada grupo

    // ...

    Juego()
    {
        // Inicializa el objeto entorno
        entorno = new Entorno(this, "Zombieland", 800, 600);


        // Inicializar lo que haga falta para el juego
        // ...


        // Inicia el juego
        entorno.iniciar();
    }

    public void tick()
    {
```

[1] **Importante:** las palabras clave "`extends InterfaceJuego`" en la definición de la clase son fundamentales para el buen funcionamiento del juego.

```
        // Procesamiento de un instante de tiempo
        // ...
    }

    public static void main(String[] args)
    {
        Juego juego = new Juego();
    }
}
```

El objeto `entorno` creado en el constructor del `Juego` recibe el juego en cuestión y mediante el método entorno.iniciar() se inicia el simulador. A partir de ahí, en cada instante de tiempo que pasa, el entorno ejecuta el método `tick()` del juego. Éste es el método más importante de la clase `Juego` y aquí el juego debe actualizar su estado interno para simular el paso del tiempo. Como mínimo se deben realizar las siguientes tareas durante cada tick:

- Actualizar el estado interno de todos los objetos involucrados en la simulación.
- Verificar si algún objeto aparece o desaparece del juego.
- Verificar si hay objetos interactuando entre sí (colisiones o superposiciones por ejemplo).
- Verificar si los usuarios están presionando alguna tecla y actuar en consecuencia (ver más abajo cómo hacer esto).
- Dibujar los mismos en la pantalla (ver más abajo cómo hacer esto). Recomendamos fuertemente que la función tick comience a dibujar durante su etapa final, luego de haber actualizado todo el estado del juego. De lo contrario, si mezclan la parte de dibujar en pantalla con la de actualizar los elementos del juego, pueden entrar en confusiones difíciles de resolver (cosas que se determina en ese tick que deben eliminarse o modificarse pero ya fueron dibujadas en pantalla).

Para dibujar en pantalla y capturar las teclas presionadas, el objeto `entorno` dispone de los siguientes métodos, entre otros:

`void dibujarRectangulo(double x, double y, double ancho, double alto, double angulo, Color color)`
↪ Dibuja un rectángulo en las coordenadas `x` e `y` de la pantalla, rotado en el ángulo dado.

`void dibujarTriangulo(double x, double y, int largo, int ancho, double angulo, Color color)`
↪ Dibuja un triángulo en las coordenadas `x` e `y` de la pantalla, rotado en el ángulo dado.

`void dibujarCirculo(double x, double y, double diametro, Color color)`
↪ Dibuja un círculo en las coordenadas `x` e `y` de la pantalla, del tamaño dado.

`void dibujarImagen(Image imagen, double x, double y, double ang)`
↪ Dibuja la `imagen` en las coordenadas `x` e `y` de la pantalla rotada en el ángulo dado.

`boolean estaPresionada(char t)`
↪ Indica si la tecla `t` está presionada por el usuario en ese momento.

`void escribirTexto(String texto, double x, double y)`
↪ Escribe el `texto` en las coordenadas `x` e `y` de la pantalla.

`void cambiarFont(String font, int tamano, Color color)`
↪ Cambia la fuente para las próximas escrituras de texto según los parámetros recibidos.

El método `estaPresionada(char t)` sirve para ver si un caracter está siendo presionado por el usuario en ese momento. Por ejemplo, `estaPresionada('A')` o `estaPresionada('+')`. Algunas teclas no pueden escribirse directamente como un `char` como por ejemplo las flechas de dirección del teclado. Para ellas, dentro de la clase entorno se encuentran las siguientes definiciones:

Valor	Tecla representada
`TECLA_ARRIBA`	Flecha hacia arriba
`TECLA_ABAJO`	Flecha hacia abajo
`TECLA_DERECHA`	Flecha hacia la derecha
`TECLA_IZQUIERDA`	Flecha hacia la izquierda
`TECLA_ENTER`	Tecla "enter"
`TECLA_ESPACIO`	Barra espaciadora
`TECLA_CTRL`	Tecla "control"
`TECLA_ALT`	Tecla "alt"
`TECLA_SHIFT`	Tecla "shift"
`TECLA_INSERT`	Tecla "ins"
`TECLA_DELETE`	Tecla "del" (o "supr")
`TECLA_INICIO`	Tecla "start" (o "inicio")
`TECLA_FIN`	Tecla "end" (o "fin")

De esta manera, para ver por ejemplo si el usuario está presionando la flecha hacia arriba se puede consultar el valor de `estaPresionada(entorno.TECLA_ARRIBA)`.

La Clase Herramientas

Ademas del Entorno, pueden utilizar la clase Herramientas que cuenta con métodos estáticos que facilitarán algunas funcionalidades. Para ver la lista de métodos de la clase Herramientas, simplemente escriban Herramientas seguido de un . (punto) en Eclipse.

Criterios de Aprobación

El producto que ustedes entregan no es solo el juego funcionando, sino también el código y el informe (principalmente estos dos últimos). Serán evaluadas principalmente las decisiones de diseño de las clases: la elección de qué clases usar, las decisiones tomadas para elegir las variables de instancia, sus métodos y los encabezados de los mismos (Nombre de las funciones bien elegidos, decisión de los datos de retorno, parámetros y tipos de los mismos). Si bien la clase Juego debe encargarse de mucha de la lógica del programa, se desea que las demás clases se hagan cargo de responsabilidades lo más posible. Cada clase debería manejar los asuntos relativos a ella. Si en algún caso esto no se puede, deberá ser aclarado en el informe. También se debe incluir en el informe cualquier decisión tomada en el desarrollo y las razones por la que se tomó esa opción y no otra. También será tenido en cuenta el trabajo en grupo, la toma de decisiones grupal y la colaboración. Recomendamos no dividir las tareas de manera

que algunos miembros del grupo se pierdan alguna parte del entendimiento. Las imágenes y sonidos en el juego no son importantes para la aprobación. No se recomienda dedicarle tiempo a estos asuntos cuando el resto del trabajo aún no esté en perfectas condiciones.

Condiciones de entrega: El trabajo práctico se debe entregar impreso y por mail a los docentes de la materia. Además del código, se debe incluir un documento en el que se describa la implementación y se detallen las decisiones tomadas durante el desarrollo. Incluir también una breve descripción del código de lós métodos no triviales. El trabajo práctico se puede hacer en grupos de hasta tres personas.

Fecha de entrega: martes 29 de octubre.