TIPS para la implementación del Trabajo Práctico PAC-MAN

- ✓ El agente PAC-MAN no puede acceder a todo el ambiente directamente ya que no lo conoce. Esto implica que la clase que representa al agente no puede tener acceso a la variable de estado del ambiente:

  celda = ambiente[x][y]; // No hacer esto

  celda = ambiente.getCelda(x,y) // No hacer esto

- ✓ La comida y el enemigo no pueden estar simultáneamente en una misma celda. El agente sí puede estar simultáneamente en una misma celda con la comida o el enemigo, pero no con ambos.
- ✓ Cuando se llama un método con parámetros de objeto, las variables que se pasan en el cuerpo del método son pasadas por referencia, lo que significa que cualquier cosa que se haga a los objetos del método, también afecta a los objetos originales. Esto incluye también a los arreglos y a todos los objetos que contienen los arreglos. Los tipos primitivos son pasados por valor.
- ✓ Consejos:
  - ➢ Clonar el objeto (generar una copia) antes de utilizarlo como argumento en un método. El método clonar, debe también clonar todos los objetos anidados dentro del objeto original.
  - ➢ Armar los árboles de búsqueda resultantes del proceso de toma de decisión del agente en cadenas anidadas entre corchetes:

    Ej: [Nodo: 1

    [Nodo: 2[Nodo: 5] [Nodo: 6]]

    [Nodo: 3[Nodo: 7]]

    [Nodo: 4]

    ]

  - ➢ O en formato XML:

```
public class Nodo implements Cloneable {
...
...
...
    public String generarXml(){

        String str = "";

        str = "<Nodo";
        str = str + " idNodo=\"" + idNodo + "\"";
        str = str + " posicion=\"(" + state.agentX + "," +""+ state.agentY +")\"";
        str = str + " orientacion=\"" + Direction.toString(state.getAgentDir()) + "\"";
        str = str + " accion=\""+ action+"\"";
        str = str + " distancia_real=\""+ realDistance+"\"";
        str = str + " habitaciones_no_visitadas=\""+ unknownRooms+"\"";
        str = str + " funcion_heuristica=\""+ heuristicFunction()+"\"";
        str = str + ">";
        str = str + toString();
        for (int i=0;i<getCantidadHijos()+1;i++) {
            str = str + hijos[i].generarXml();
        }
        str = str + "</Nodo>";

        return str;

    }
```

- ✓ Links de interés:
  - Blog de la materia: http://iacatedra.blogspot.com
  - Artificial Intelligence, A Modern Approach: http://aima.cs.berkeley.edu/
  - Search and game playing: http://aima.cs.berkeley.edu/ai.html#search
  - Ejemplos y estructuras de datos: http://www.java2s.com/
  - Java Programming Style Guidelines: http://geosoft.no/development/javastyle.html
  - Robocup: http://www.robocup.org/

Interfaz de la clase "Calculador

Constructores:

public Calculador(String nombreGrupo)
Genera una instancia de la clase calculador y asigna el parámetro *nombreGrupo* como nombre del grupo. Este nombre será utilizado en el simulador para identificar al grupo.

public Calculador()
Genera una instancia de la clase calculador sin asignar un nombre al grupo.

Métodos de instancia:

public int calcularEnergiaPacMan()
Retorna un entero que indica la energía inicial del agente Pac-Man.

public int calcularEnergiaPacMan(String accion)
Retorna un entero que indica la energía actual del agente Pac-Man luego de haber ejecutado la acción *accion* pasada como parámetro. Las acciones permitidas son "arriba", "izquierda", "abajo", "derecha", "comer" y "pelear".

public int getPerformance()
Retorna un entero que indica la performance que obtuvo el agente Pac-Man. Éste método sólo puede ser ejecutado una vez.

public Pair getPosicionInicial()
Retorna un Pair (fila, columna) que representa la fila y columna en la cual se encuentra el agente Pac-Man.

public Vector inicializarComida()
Retorna un vector que contiene elementos de tipo Pair (fila, columna). Estos pares representan las posiciones iniciales en las que se encuentra la comida.

public Vector inicializarEnemigo()
Retorna un vector que contiene elementos de tipo Pair (fila, columna). Estos pares representan las posiciones iniciales en las que se encuentran los enemigos.

Ayuda para el uso del "calculador.jar"

Cómo agregar un archivo .jar a un proyecto en Eclipse.

Se debe tener un Proyecto Java abierto. Para importar el archivo *calculador.jar*, ir al menú *Proyect→Properties* como muestra la figura 1.

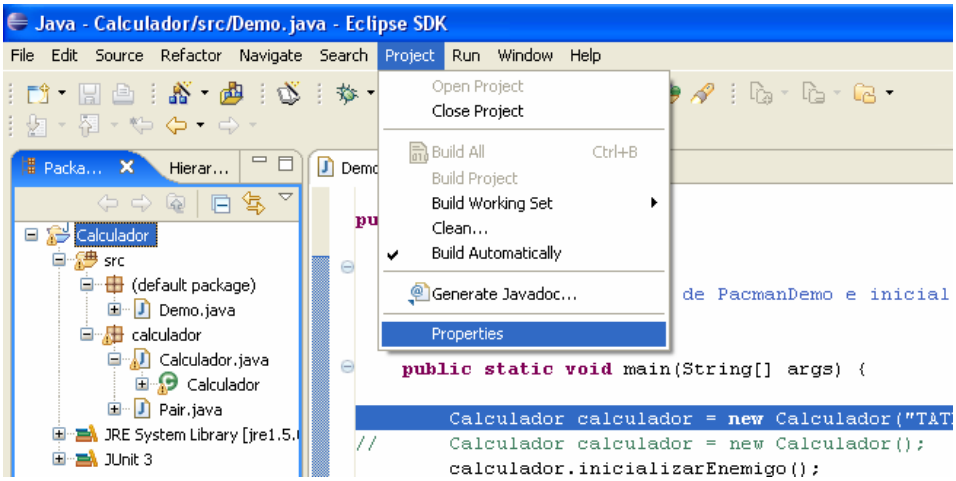

Figura 1: propiedades de un proyecto en Eclipse.

Luego en la sección "*Java Build Path*" hacer clic en "*Add External JARs*" y seleccionar la ubicación del archivo (Figura 2). Hacer clic en *Ok* y listo.

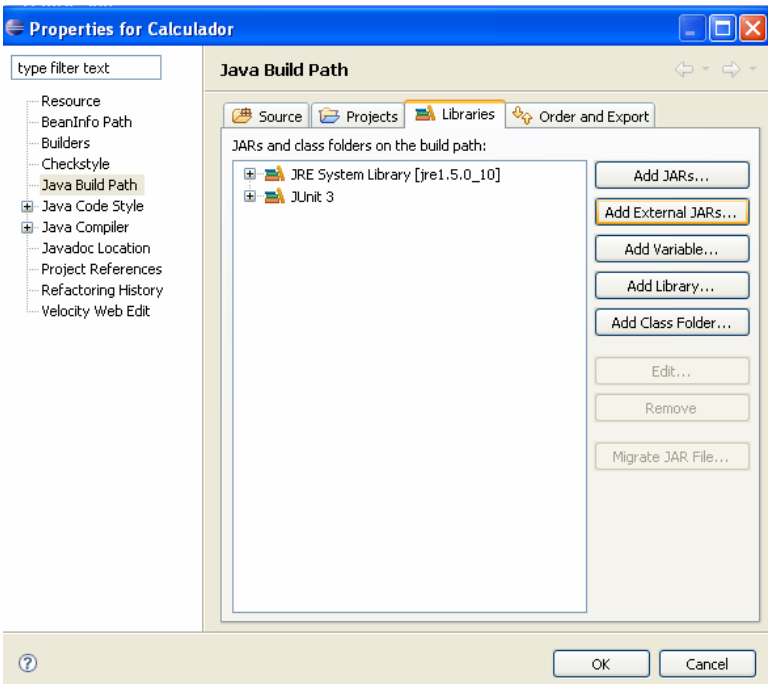

Figura 2: cómo agregar un archivo .jar a un proyecto en Eclipse.

Cómo generar una instancia de la clase Calculador:

La siguiente línea de código permite generar una instancia de la clase Calculador.

```
Calculador calculador = new Calculador("TATE-Pac");
```

La cadena pasada como parámetro representa el nombre asignado al grupo que será mostrado en la simulación.

Cómo inicializar el mundo del agente Pac-Man:

Una vez generada la instancia se debe inicializar el mundo en el que se desenvuelve el agente Pac-Man. Para lograr esto son necesarias las siguientes líneas de código:

```
Vector enemigos = calculador.inicializarEnemigo();
Vector comida = calculador.inicializarComida();
```

Las variables *enemigos* y *comida* son vectores que contienen elementos de tipo Pair (fila, columna) que representan las posiciones donde se encuentran los enemigos y la comida.

IMPORTANTE!! Los elementos de tipo Pair están compuestos por (filas, columnas), por lo tanto el método par.x() representa la fila y el método par.y() representa la columna. Las coordenadas de la grilla se muestran en la siguiente tabla:

1,1	1,2	1,3	1,4
2,1	2,2	2,3	2,4
3,1	3,2	3,3	3,4
4,1	4,2	4,3	4,4

Cómo obtener la posición inicial del agente Pac-Man:

Para obtener la posición inicial se debe ejecutar el método getPosicionInicial():

```
Pair pos = calculador.getPosicionInicial();
```

Este método retorna un par con la posición inicial del agente Pac-Man. Otra opción puede ser la siguiente:

```
int posFila = calculador.getPosicionInicial().x();
int posCol = calculador.getPosicionInicial().y();
```

Cómo obtener la energía inicial del agente Pac-Man:

Para obtener la energía inicial se debe ejecutar el método calcularEnergiaPacman() sin parámetros:

```
int energia = calculador.calcularEnergiaPacMan();
```

Este método retorna un entero con la energía inicial del agente Pac-Man.

Cómo obtener la energía actual del agente Pac-Man:

Luego de ejecutar una acción es posible que la energía del agente cambie. La siguiente línea de código muestra como obtener la energía actual del Pac-Man:

```
int energia = calculador.calcularEnergiaPacMan("pelear");
```

Este método debe ser invocado luego que la estrategia de búsqueda haya determinado la acción más adecuada para la situación actual.

Cómo obtener la performance del agente Pac-Man:

La performance debe ser obtenida al finalizar la ejecución del agente Pac-Man:

```
int performance = calculador.getPerformance();
```

Este método puede ser ejecutado sólo una vez mientras dure la instancia de la clase Calculador. En caso de ser ejecutado más de una vez retornará -1.