Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

IIC1103 – Introducción a la Programación
2 - 2017

Enunciado Tarea 2

Recordatorio:

- **Fecha de entrega:** 19 de Octubre de 2017, a las 23:50 hrs.
- **Foro de consulta:** `https://piazza.com/uc.cl/summer2017/iic1103`
- Este trabajo es **estrictamente personal**. Recuerda leer la Política de Integridad Académica del DCC disponible en `http://www.ing.uc.cl/ciencia-de-la-computacion/programas/licenciatura/politica-de-integridad-academica/`. Se usará un software **anti-plagio** para detectar similitud entre códigos.

¡Atención!

Ten en consideración que **no se recibirán entregas fuera del plazo**. Tampoco se evaluará tu tarea si te equivocas al entregar el archivo.
Es tu responsabilidad subir entregas parciales de tu tarea. Se revisará **solamente la última versión** que hayas subido. Revisa la sección **Entrega** para las instrucciones de entrega de tu trabajo.

Objetivo

En esta tarea se espera que apliques los contenidos de listas, listas de listas, archivos y manejo de *strings*.

Enunciado

En esta tarea tendrás que programar el funcionamiento del juego "El coyote y las gallinas" o "Cercar la liebre". Este juego consiste en un tablero de 5×5 en el cual se enfrentan dos jugadores. Al iniciar el juego, se posicionan 12 gallinas y 1 coyote de la forma indicada en la Figura 1. Un jugador controla a las gallinas, moviendo solamente una en cada turno, mientras que el otro jugador controla al coyote. Si deseas probarlo puedes hacerlo en el siguiente link: `http://www.lutanho.net/play/coyoteandchickens.html`.

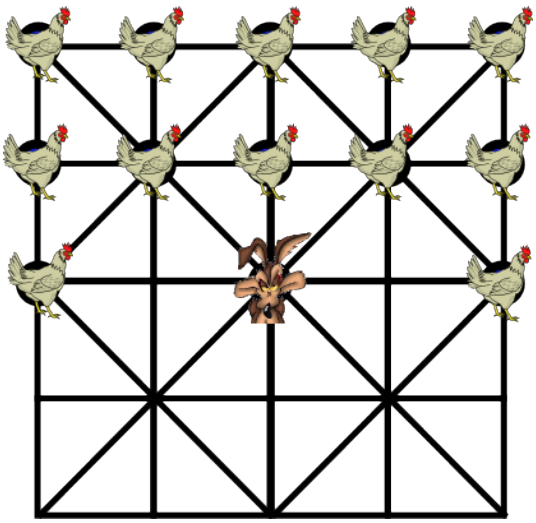

Figura 1: Tablero Inicial

Objetivo del juego

Objetivo de las gallinas: Rodear al coyote e impedirle que pueda ejecutar una jugada. Tomando como ejemplo la Figura 2, el coyote no puede hacer ningún movimiento ya que en todas las direcciones hay al menos dos gallinas.

Objetivo del coyote: Eliminar el mayor número posible de gallinas para que no puedan atraparle, lo cual se cumple cuando el coyote ha comido 2 o más gallinas. En la Figura 3, el coyote ha comido exactamente 2 gallinas durante la partida, por lo tanto éste gana con seguridad.

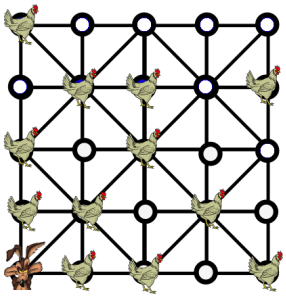

Figura 2: Ejemplo donde las gallinas han ganado

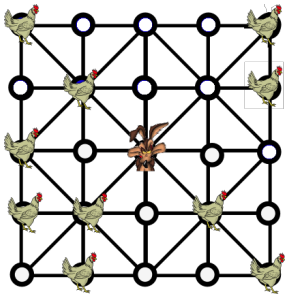

Figura 3: Ejemplo donde el coyote ha ganado

Reglas del juego

Inicio

- Se colocan las fichas en la posición inicial, según la Figura 1.
- Las gallinas siempre comienzan la partida.

Desarrollo

- Los jugadores, por turno, mueven **una** de sus piezas a una casilla vacía contigua que esté conectada según la estructura del tablero.
- El coyote puede *comer* a una gallina saltando sobre ella (como en las damas). La gallina debe estar en un espacio conectado directamente a la posición del coyote y tener un espacio vacío detrás de ella.
- El coyote puede comer varias gallinas en un solo turno por medio de varios saltos. Cada vez que el coyote ha comido una gallina, si aún puede comer otra gallina a partir de su nueva posición, entonces **debe** volver a mover para saltar sobre otra gallina en el mismo turno.
- Las gallinas **no** se pueden comer al coyote. Su único objetivo es dejarlo encerrado.

Fin del juego

El juego termina cuando las gallinas o el coyote cumplen su objetivo.

Estructura del Tablero

El tablero que utiliza este juego se llama alquerque.[1] La estructura de este tipo de tablero no es convencional, pues una ficha no puede moverse en cualquier dirección. Solamente está permitido moverse a espacios conectados entre sí, tal como lo indican las líneas del mismo tablero. Cabe agregar que el coyote no puede ir a un espacio ocupado por otra ficha y las gallinas tampoco. En la Figura 4 se pueden ver las jugadas válidas que el coyote puede realizar dada la posición de ejemplo.

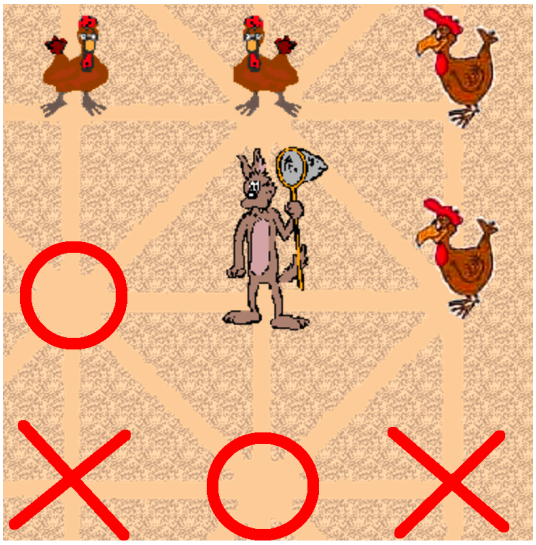

Figura 4: Movimientos válidos para el coyote representados con círculos

Tu programa debe respetar este movimiento. Por lo tanto, si un jugador intenta hacer una jugada inválida, tu programa debe avisar que no puede realizarla y pedir que ingrese una nueva jugada. Ningún jugador puede saltar su turno.

[1] ¿No lo sabía? En este curso siempre se aprende algo nuevo.

El sistema de coordenadas se detalla en la Figura 5.

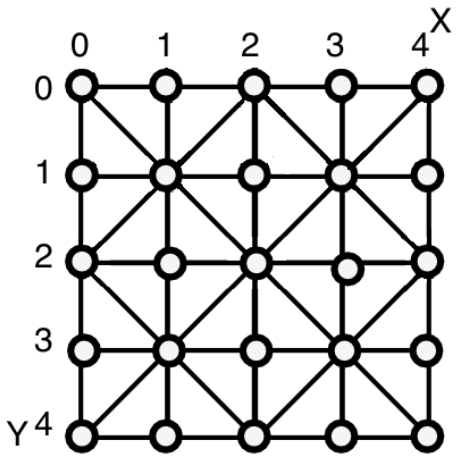

Figura 5: Sistema de coordenadas del tablero

Interacción con el usuario

El programa debe permitir guardar y recuperar partidas usando archivos. Al ejecutar tu programa debes preguntar al usuario si desea cargar una partida. Si responde afirmativamente entonces debes preguntar el nombre del archivo y cargar dicha partida tal como se detallará en la sección **Archivos**. En caso contrario debes preguntar el nombre de cada jugador y qué papel tomará: ser las gallinas o el coyote. No se permite que ambos jugadores sean el coyote, ni tampoco que ambos sean las gallinas, en este caso hay que volver a preguntar a los jugadores qué quieren ser cada uno. Al comenzar una partida nueva, el tablero generado siempre corresponde al especificado en la Figura 1.

Durante cada turno, y antes de hacer la jugada, debes preguntar al usuario si desea (1) guardar el estado de la partida y salir del juego; (2) salir sin guardar; o (3) seguir jugando. Si elige seguir jugando, entonces debes preguntar al jugador desde y hacia dónde quiere mover su ficha. El jugador debe indicar el movimiento deseado de acuerdo al siguiente formato:

```
pos_x_inicial,pos_y_inicial,pos_x_final,pos_y_final
```

En caso que el jugador no pueda mover una pieza a cierto lugar, ya sea porque esa posición está ocupada o la estructura del tablero no lo permite, entonces debes avisarle al usuario y preguntar de nuevo por el movimiento.

En caso de que el coyote se coma una gallina y aún exista la posibilidad de comer otra, entonces el coyote debe jugar de nuevo de tal forma que se coma la siguiente gallina. En este caso debes mostrar el nuevo estado del tablero, pedir las coordenadas del movimiento al jugador y prohibir hacer un movimiento que no sea para comer una gallina. A pesar de que el coyote haga más de un movimiento, esto se considera como un solo turno, por lo tanto, debes revisar si hay ganador solamente cuando el coyote haya terminado de comer todas las gallinas posibles. No debe ser posible guardar la partida hasta que el coyote termine de comer y termine su turno.

Luego de que el jugador haya terminado un movimiento, debes mostrar el nuevo estado del tablero. Para mostrar a las gallinas simplemente imprime una letra `G` en cada posición en la cual se encuentren, mientras que el coyote será representado con la letra `C`, los espacios vacíos se muestran con una `N`. He aquí un ejemplo de interacción:

```
¡Bienvenido al juego del coyote y las gallinas!
¿Quieres cargar una partida (1) o empezar de nuevo (2)?
> 2
¡Perfecto! Dime el nombre del jugador que será el coyote:
> Pedro
Dime el nombre del jugador que será las gallinas:
> Carlito
¡Comencemos!
G-G-G-G-G
|\|/|\|/|
G-G-G-G-G
|/|\|/|\|
G-N-C-N-G
|\|/|\|/|
N-N-N-N-N
|/|\|/|\|
N-N-N-N-N

Es tu turno Carlito, ¿Cuál es tu movimiento?
¿o deseas guardar la partida y salir (G)?
¿o simplemente salir (S)?
> 4,2,4,3
G-G-G-G-G
|\|/|\|/|
G-G-G-G-G
|/|\|/|\|
G-N-C-N-N
|\|/|\|/|
N-N-N-N-G
|/|\|/|\|
N-N-N-N-N

Es tu turno Pedro, ¿Cuál es tu movimiento?
¿o deseas guardar la partida y salir (-1)?
¿o simplemente salir (-2)?
.
.
.
--------------------------------

¡Bienvenido al juego del coyote y las gallinas!
¿Quieres cargar una partida (1) o empezar de nuevo (2)?
> 1
¿Cuál es el nombre del archivo?
> juego_1.txt
```

En caso de cargar la partida se debe mostrar el historial de las jugadas, esto quiere decir que cada movimiento debes reproducirlo como si cada jugador hubiera jugado normalmente. A partir de aquí debe ser posible continuar jugando sobre esa misma partida.

Al momento de guardar la partida debes escribir un archivo `.txt` de acuerdo al formato que se presenta en la sección **Archivos**. Debes preguntar al usuario por el nombre que desea ponerle al archivo.

Por último, al final de cada turno debes comprobar si existe un ganador, tal como se detalla en la sección **Objetivo del juego**. En caso de que haya ganador, debes comunicar por consola quién es el ganador, preguntar si desean guardar esta partida, y luego si desean jugar de nuevo, cargar una partida o dejar de jugar.

Archivos

Deberás ser capaz de leer y escribir archivos que registren la lista de movimientos de una partida. Cada archivo tiene el siguiente formato:

```
NombreJugadorCoyote,NombreJugadorGallinas
rol,pos_x_inicial,pos_y_inicial,pos_x_final,pos_y_final
rol,pos_x_inicial,pos_y_inicial,pos_x_final,pos_y_final
.
.
.
rol,pos_x_inicial,pos_y_inicial,pos_x_final,pos_y_final
```

Cada línea corresponde a un movimiento, excepto la que muestra los nombres de los jugadores. Cuando el programa cargue una partida debe mostrar en consola cada movimiento del archivo. En caso de que el coyote se haya comido más de una gallina en su turno es necesario detallar cada movimiento, mostrando cómo va cambiando el tablero con cada jugada.

A continuación se explican los parámetros del formato descrito:

- `NombreJugadorCoyote`: Nombre del jugador que juega como coyote.
- `NombreJugadorGallinas`: Nombre del jugador que juega como las gallinas.
- `rol`: será representado como C en caso de que el jugador sea el coyote, y G cuando juegan las gallinas.
- `pos_x_inicial` y `pos_y_inicial`: posición inicial de la ficha a mover.
- `pos_x_final` y `pos_y_final`: posición final de la ficha a mover.

El siguiente archivo se muestra a modo de ejemplo:

```
Pedro,Carlito
G,3,1,3,2
C,2,2,2,3
G,1,1,1,2
C,2,3,3,3
G,1,2,1,3
C,3,3,3,1
C,3,1,1,1
```

Para probar la lectura de archivos de tu programa puedes utilizar el ejemplo antes descrito. Luego de cargar dicho archivo y mostrar el historial de jugadas, debiese quedar el siguiente tablero en consola.

```
G-G-G-G-G
|\|/|\|/|
G-C-N-N-G
|/|\|/|\|
G-N-N-N-G
|\|/|\|/|
N-G-N-N-N
|/|\|/|\|
N-N-N-N-N
```

Normalmente, después de imprimir el historial y el estado final del tablero, debes dar la posibilidad de seguir jugando. Pero en este ejemplo el coyote logra comer 2 gallinas, de modo que gana la partida. En este caso debes imprimir al ganador y preguntar si desean guardar esta partida, y luego si desean jugar de nuevo, cargar otra partida o dejar de jugar.

Se te entregará un archivo de extensión `.txt` con un ejemplo de carga de una partida, donde se muestra la salida de consola correspondiente.

Puedes suponer que el archivo aquí mostrado está correcto.

Entrega

Debes guardar tu tarea en un archivo de nombre `tarea2_numero_alumno.py`, donde debes reemplazar `numero_alumno` con tu número de alumno. Por ejemplo, si tu número de alumno es 12345678 el nombre de la tarea sería `tarea2_12345678.py`. **Si no sigues estas instrucciones de entrega, el ayudante corrector tiene el deber de descontar 5 décimas a tu tarea.**

La entrega se realiza a través de un cuestionario en el SIDING disponible en la página web del curso hasta el 19 de Octubre de 2017 a las 23:50 hrs. **No se recibirán entregas atrasadas ni entregadas por otro medio.**

Indicaciones generales

1. Recuerda que la tarea es **estrictamente individual**. Cualquier situación de copia será sancionada severamente según dicta el código de honor de la universidad, el cual se detalla al final de este documento.
2. **NO hagas la tarea a última hora**, pues el cuestionario se cierra automáticamente cuando se cumple el plazo. **NO se aceptarán entregas atrasadas ni entregadas por otros medios.**
3. Revisa bien lo que entregas, y prueba tu código antes de enviar la versión definitiva. **Puedes enviar tu tarea todas las veces que estimes conveniente, pero solo se almacenará y revisará la última entrega**. Si el ayudante corrector no logra hacer correr tu tarea, es bastante probable que ésta sea evaluada con una nota baja o cercana a 1,0.

Política de Integridad Académica

Los alumnos de la Escuela de Ingeniería deben mantener un comportamiento acorde al Código de Honor de la Universidad:

"Como miembro de la comunidad de la Pontificia Universidad Católica de Chile me comprometo a respetar los principios y normativas que la rigen. Asimismo, prometo actuar con rectitud y honestidad en las relaciones con los demás integrantes de la comunidad y en la realización de todo trabajo, particularmente en aquellas actividades vinculadas a la docencia, el aprendizaje y la creación, difusión y transferencia del conocimiento. Además, velaré por la integridad de las personas y cuidaré los bienes de la Universidad."

En particular, se espera que mantengan altos estándares de honestidad académica. Cualquier acto deshonesto o fraude académico está prohibido; los alumnos que incurran en este tipo de acciones se exponen a un procedimiento sumario. Ejemplos de actos deshonestos son la copia, el uso de material o equipos no permitidos en las evaluaciones, el plagio, o la falsificación de identidad, entre otros. Específicamente, para los cursos del Departamento de Ciencia de la Computación, rige obligatoriamente la siguiente política de integridad académica en relación a copia y plagio: Todo trabajo presentado por un alumno (grupo) para los efectos de la evaluación de un curso debe ser hecho individualmente por el alumno (grupo), sin apoyo en material de terceros. Si un alumno (grupo) copia un trabajo, se le calificará con nota 1.0 en dicha evaluación y dependiendo de la gravedad de sus acciones podrá tener un 1.0 en todo ese ítem de evaluaciones o un 1.1 en el curso. Además, los antecedentes serán enviados a la Dirección de Docencia de la Escuela de Ingeniería para evaluar posteriores sanciones en conjunto con la Universidad, las que pueden incluir un procedimiento sumario. Por "copia" o "plagio" se entiende incluir en el trabajo presentado como propio, partes desarrolladas por otra persona. Está permitido usar material disponible públicamente, por ejemplo, libros o contenidos tomados de Internet, siempre y cuando se incluya la cita correspondiente.