Algoritmos y Programación I
(75.40/95.14)

Trabajo práctico nº1

Primer cuatrimestre 2016

Introducción

Hace unos días recibimos un extraño mensaje de voz. El archivo estaba dañado, pero luego de un arduo proceso de recuperación logramos extraer los siguientes fragmentos:

> Fecha *∗ruido∗* Hace años, aquel fatídico 10 de marzo de 2016, el programa *AlphaGo* vuelve a ganarle una partida de *Go* al mejor jugador del mundo[1]. En *∗ruido∗* no fuimos capaces de ver que se trataba del principio del fin. Las máquinas *∗ruido∗* completamente el control *∗ruido∗*. Nuestra única esperanza es ganar una partida de *Othello* *∗ruido∗* prometieron liberarnos si ganábamos.
>
> Debemos entrenar y no tenemos cómo *∗ruido∗*. Necesitamos un simulador de *Othello*, sólo sabemos programar en Pascal y es imposible para nosotros *∗ruido∗*. Si no podemos vencerlas *∗ruido∗* fin de la humanidad *∗ruido∗* última esperanza *∗ruido∗* Python.

Es entonces que esta difícil tarea es encomendada a un prometedor grupo de programadores de la FIUBA. Pesa ahora sobre sus hombros, no solo la aprobación de un trabajo práctico, sino tambien el futuro de la humanidad.

[1] http://www.theguardian.com/technology/2016/mar/10/google-alphago-ai-wins-second-game-against-go-champion-lee-sedol

Consigna

Se pide implementar un juego interactivo de *Othello*, también conocido como *Reversi*[2].

Reglas del juego

- El juego se desarrolla en un tablero cuadrado de 8×8 entre dos jugadores: blanco y negro.
- El juego se basa en capturar fichas del jugador oponente y transformarlas a fichas propias. El jugador con más fichas al finalizar el juego es el ganador.
- Al iniciar el juego el tablero esta completamente vacío salvo por las primeras 4 fichas en el centro del mismo, que se colocan de la forma:

```
B N
N B
```

- El juego se desarrolla por turnos. El jugador negro tiene el primer turno.
- En su turno, cada jugador debe colocar una pieza en el tablero, a menos que no sea posible, en cuyo caso debe pasar el turno.
- El juego termina cuando ninguno de los dos jugadores tiene un movimiento posible.
- Para simplficar, el juego tendrá un maximo de 60 turnos. Al terminarse esos turnos se decide un ganador aun cuando queden casilleros vacíos o jugadas posibles.

Movimientos válidos

- Un movimiento consiste en ubicar una pieza en un casillero vacío, en una posición que permita capturar una o más piezas del oponente.
- Cuando una ficha es capturada, se reemplaza por una ficha del color opuesto.

[2] https://es.wikipedia.org/wiki/Reversi

- Una fila de una o más fichas es capturada cuando está rodeada en ambos extremos por dos fichas del oponente, y una de las dos fichas es la que se acaba de colocar en el tablero.
- Una ficha puede capturar cualquier número de fichas oponentes en una o más filas, en cualquier dirección (horizontal, vertical, diagonal).

Por ejemplo, si es el turno del jugador blanco y ubica una ficha en el casillero `X`, se capturan las fichas negras y el resultado se muestra a la derecha:

```
B              B
NNB            BNB
N B    --->    B B
NN             BB
X              B
```

Notar que queda una ficha negra que no es capturada, incluso estando rodeada por dos fichas blancas. Sólo la ficha blanca que se ubicó en el casillero X puede capturar.

Requerimientos

- El juego debe tener por defecto un tablero de 8×8, pero debe poder ser configurable sin tener que re-escribir todo el codigo.
- Al mostrar el tablero en la pantalla, se debe marcar las filas y columnas de alguna manera que permita al jugador indicar fácilmente su próxima jugada. Se recomienda usar una distinta denominación para las columnas y las filas; por ejemplo $A3$ puede representar la columna A y la fila 3.

Criterios de aprobación

A continuación se describen los criterios y lineamientos que deben respetarse en el desarrollo del trabajo.

Informe

El informe debe consistir en una descripción del **diseño** del programa.

Debe recordarse que la etapa de diseño es *anterior a la implementación*, por lo tanto debe describirse, utilizando texto y/o diagramas, cómo se va a estructurar el código para cumplir con las especificaciones de la consigna.

Algunas preguntas que deberían responderse:

- A grandes rasgos, ¿cómo será el flujo del programa?
- ¿Cómo se va a guardar en memoria el estado del juego?
- ¿Qué operaciones se efectuarán durante el juego?

Código

Además de satisfacer las especificaciones de la consigna, el código entregado debe cumplir los siguientes requerimientos:

- El código debe ser claro y legible.
- El código debe estructurarse en funciones y, cuando corresponda, módulos. Las funciones deben definirse de la manera más genérica posible.
- Todas las funciones deben estar adecuadamente documentadas, y donde sea necesario el código debe estar acompañado de comentarios.

Entrega

La entrega del trabajo consiste en:

- El informe y código fuente impresos. Para el código fuente utilizar una tipografía `monoespacio`.
- El informe en formato *PDF*.
- Una versión digital de todos archivos `.py` de código, separados del informe. En el caso de ser más de un archivo, comprimidos en un `.zip`.

El informe impreso debe entregarse en clase. Los dos últimos (PDF y código fuente) deben enviarse a la dirección electrónica `tps.7540rw@gmail.com` con el asunto "TP1 - <*Padrón*>".

Este trabajo práctico se desarrolla en forma **individual**. El plazo de entrega vence el **lunes 18 de abril de 2016**.