



LÓGICA PARA CIENCIAS DE LA COMPUTACIÓN

Proyecto N° 1

GO

Primer Cuatrimestre de 2019

El juego

GO es un juego de mesa para dos jugadores que consiste de un tablero de 19 líneas horizontales (filas) por 19 líneas verticales (columnas) y piezas llamadas *pedras* (fichas) de color blanco y negro (ver figura 1).

El tablero se encuentra vacío al comienzo del juego, y los jugadores, llamados *blanco* y *negro*, deben colocar pedras del color correspondiente, por turnos, en las *intersecciones* de las líneas verticales y horizontales. El jugador negro realiza la primer movida, luego de lo cual blanco y negro alternan. Una movida consiste en ubicar una piedra del color propio en una intersección vacía del tablero. Una vez que una piedra es ubicada ya no puede moverse, pero puede ser capturada como resultado de una jugada del jugador contrario. Al colocar una piedra, todas aquellas pedras del oponente que queden *rodeadas* (o *encerradas*) por pedras propias serán capturadas, esto es, removidas del tablero.

Captura de pedras

Dada una configuración del tablero, decimos que un conjunto dado A (no vacío) de pedras de color C (blanco o negro) es un conjunto de pedras rodeadas si y solo si cada piedra de A verifica que las posiciones adyacentes, vertical u horizontalmente, que la circundan contienen pedras del color contrario, o pedras de color C que también pertenecen a A (es decir, que también están rodeadas). Cuando un jugador coloca una piedra, todas aquellas pedras del oponente que quedan rodeadas (si es que alguna) serán capturadas.

OBSERVACIONES:

- dos posiciones del tablero son adyacentes si coinciden en la fila y se encuentran en columnas contiguas (adyacentes horizontales), o coinciden en la columna y se encuentran en filas contiguas (adyacentes verticales).
- luego de colocar una piedra, podría existir más de un conjunto de pedras del oponente rodeadas, y todas deberán ser capturadas.

Jugadas válidas

Diremos que colocar una piedra de color C en una intersección dada constituiría *suicidio* si como resultado de colocarla en dicha posición se generaría un conjunto de pedras rodeadas de color C (por supuesto, entre las cuales estará la piedra colocada), salvo que pedras contrarias también queden atrapadas, en cuyo caso no estaríamos en presencia de suicidio. Un jugador podrá colocar una piedra en cualquier intersección vacía, con la restricción de que al colocarla

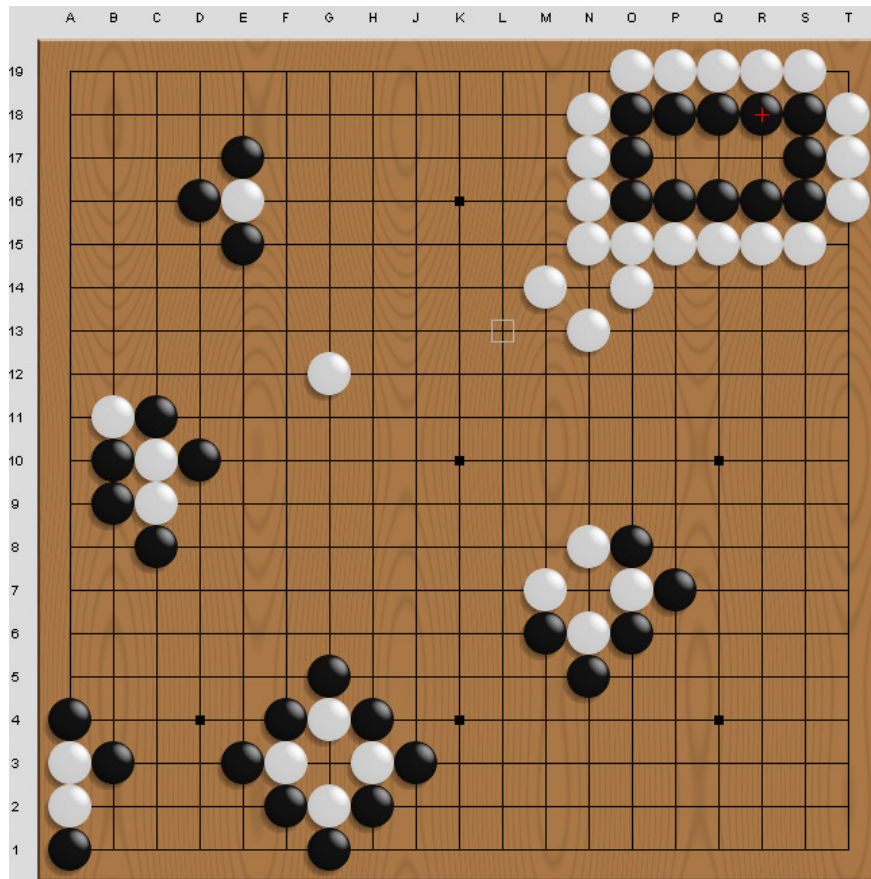


Figura 1: Tablero y posible configuración del juego

en dicho lugar no cometa suicidio. Además de esto, un jugador puede *pasar* (saltar) su turno en cualquier momento.

Finalización del juego y Puntaje

El juego termina cuando los dos jugadores pasaron consecutivamente. El puntaje de cada jugador es el número de intersecciones que quedaron ocupadas por piedras propias, además de aquellas intersecciones vacías que quedaron encerradas o *rodeadas* por piedras propias (*territorio*). El jugador con mayor puntaje es el ganador. OBSERVACIÓN: la definición de conjunto de intersecciones vacías rodeadas por piedras de un jugador es análoga a la que se usa para determinar la captura de piedras contrarias, especificada anteriormente.

Tutorial On-Line

Para mayor información acerca del GO se recomienda consultar el tutorial interactivo *The Interactive Way To Go*.

Requerimientos del Proyecto

A continuación se especifican los requerimientos para el presente proyecto.

Funcionalidad

Debe implementarse una aplicación que permita jugar GO a dos usuarios. La aplicación debe ofrecer una interfaz gráfica apropiada, al estilo de las versiones disponibles en línea, satisfaciendo los siguientes requerimientos:

1. permitir a los jugadores efectuar movidas (incluyendo pasar su turno), mostrando la evolución de la partida, principalmente la configuración del tablero en cada momento.
2. avisar al usuario si éste intenta realizar una jugada inválida (además de no permitirla).
3. detectar la finalización del juego (ambos jugadores pasan consecutivamente) mostrando el puntaje de cada jugador e indicando el ganador.
4. [opcional] implementar la modalidad de juego contra la máquina, donde un jugador se enfrenta a la *computadora* en el juego. La inteligencia de la computadora deberá mínimamente detectar jugadas de captura, ya sea propias (para efectuarlas y capturar piedras del jugador humano) o del humano (para intentar evitarlas). Cuando no existan jugadas de captura que realizar y/o evitar, deberá jugar intentando armar configuraciones tendientes a conseguir (o evitar) capturas. Mínimamente debe tratar de dejar piedras del oponente en atari.

Implementación

La funcionalidad descrita previamente debe ser implementada mediante una aplicación web.

- La interfaz gráfica del juego consistirá de una interfaz web implementada empleando los tres lenguajes básicos de desarrollo web del lado del cliente: HTML, CSS y JAVASCRIPT.
- La lógica detrás del juego deberá ser implementada en PROLOG, esto es, la representación de las configuraciones de la grilla, el cambio de una configuración a otra consecuencia de *aplicar* un color, y los cálculos detrás de las funcionalidades de ayuda. Dicha lógica en PROLOG correrá del lado del servidor, en un servidor web Pengines (PROLOG engines).

Importante: cualquier duda acerca de si se debe implementar una determinada funcionalidad del lado del cliente o del servidor (PROLOG), consultar con la cátedra.

Documentación

Se deberá realizar un informe que explique claramente la implementación en PROLOG realizada, además de indicar qué predicados son exportados para ser consultados desde la interfaz web en el cliente y para qué funcionalidad de la interfaz son utilizados. **Aclaración:** No es necesario explicar la implementación de la interfaz web más allá de indicar a alto nivel cómo se relaciona con la implementación en PROLOG. Además, deberá escribirse una sección que explique brevemente los pasos requeridos para interactuar con la interfaz web implementada.

Se recomienda estructurar el informe de manera top-down, comenzando con una descripción a alto nivel de la implementación. Puede aprovechar el informe para destacar características positivas de la resolución, y documentar cualquier otra observación que considere pertinente.

Importante: en el desarrollo de software, la documentación de la implementación constituye un elemento fundamental. Es por esto que, para la evaluación del presente proyecto, se dará suma importancia a la calidad (claridad y completitud) del informe entregado. Aproveche los horarios de práctica para consultar acerca del desarrollo del informe.

Comisiones y Entrega

1. Las comisiones pueden estar conformadas por hasta 2 integrantes, y deben ser registradas en la página de la materia.
2. La fecha límite de entrega del presente proyecto es el día viernes 17 de Mayo a las 23:59. Los proyectos entregados fuera de término recibirán una penalización en su calificación, la cual será proporcional al retraso incurrido.
3. La entrega del proyecto consiste del envío por mail de la resolución del proyecto y versión electrónica del informe.
 - Enviar por mail directamente al integrante de cátedra asignado a la comisión, con copia al asistente (en caso de no ser el asignado). Mails:
 - Damián: `damian.fernando.varela+LCC@gmail.com`
 - Federico: `fschmidt.10@hotmail.com`
 - Noni: `ac@cs.uns.edu.ar`
 - Mauro (asistente): `mgomezlucero+LCC@gmail.com`
 - Asunto del mail: “Proyecto LCC - Comisión <Ap.y Nom. Integrantes>”
 - Adjunto: un .zip conteniendo 1) la carpeta ‘proylcc’ (`pengines-master/apps/proylcc`) de la aplicación en Pengines y 2) un pdf con la versión electrónica del informe.