

Bloque 1 – Práctica 1

En este trabajo vamos a poner en práctica conocimientos relacionados con procesos, signals y pipes. Es posible, aunque no absolutamente necesario, que necesitéis threads.

A partir de aquí, vamos a ver una idea general de lo que hay que hacer y lo vamos a ir concretando poco a poco y punto por punto.

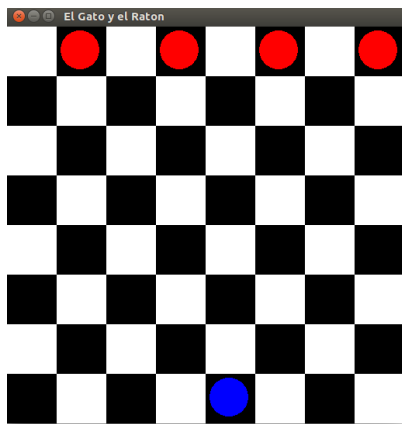
Ítem 1: Idea general de lo que hay que hacer

Vamos a implementar el juego de “El gato y el ratón”.

Es un juego de tablero y los elementos del juego son:

- Un tablero de ajedrez
- 1 peón blanco (para el ratón)
- 4 peones negros (para el gato)

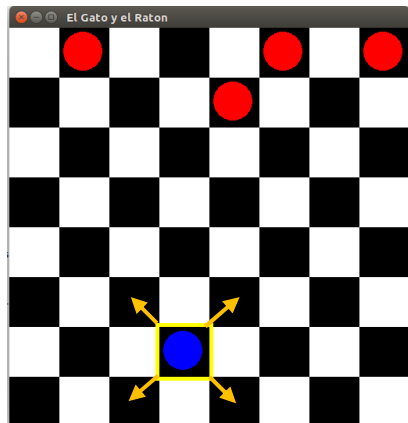
La colocación inicial de las piezas es la siguiente (estamos suponiendo que la azul es el ratón y las rojas el gato):



¿Cómo se mueve el ratón?

Se mueve en las 4 diagonales, pudiendo desplazarse sólo una casilla por turno.

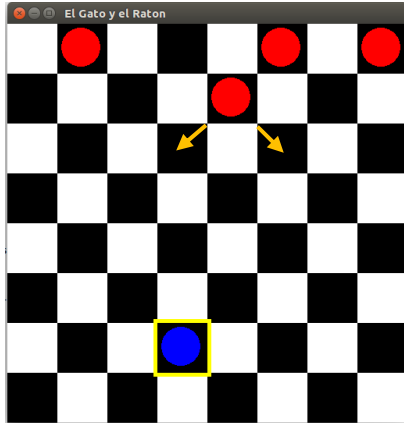
No puede moverse a una casilla que está ocupada por una pieza del gato



¿Cómo se mueve el gato?

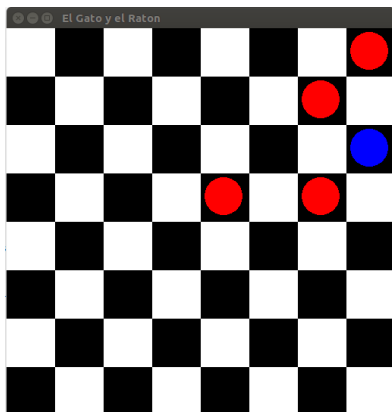
Puede mover una pieza por turno.

La pieza que decide mover sólo puede avanzar una casilla hacia adelante y en las diagonales, siempre que la casilla destino no esté ocupada por otra pieza, bien sea del mismo gato o del ratón.



¿Cuál es el objetivo del gato?

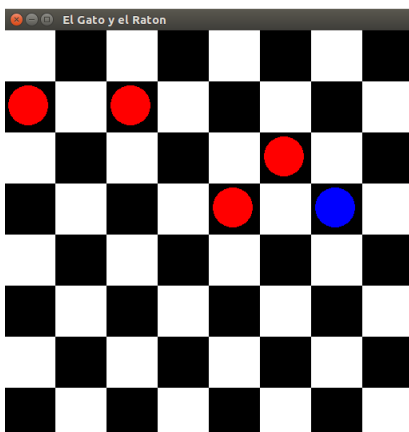
El objetivo del gato es acorralar al ratón, dejándolo en una posición en la que no se pueda mover.



¿Cuál es el objetivo del ratón?

Escapar al gato. Llegar hasta el otro lado del tablero sin que el gato lo pueda acorralar.

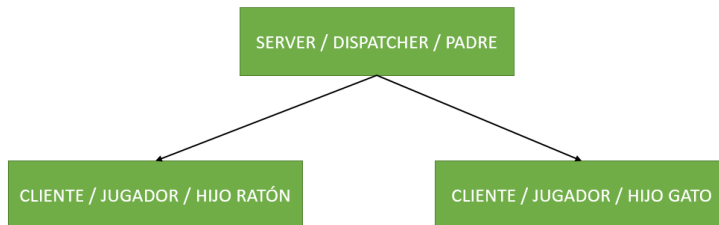
En este caso, ya se ve que ha ganado el ratón, aunque no haya llegado hasta el final del tablero.



Ítem 2: Organización en procesos

En este aspecto, al ser un juego tan sencillo, existen más de una alternativa válida y que sería funcional al 100%.

Sin embargo, pensando en la siguiente práctica, se os propone la siguiente:



Vamos a ver lo que se espera de cada uno procesos.

Ítem 3: Proceso padre

Se espera lo siguiente:

1. Debe crear las vías de comunicación con los hijos
2. Crea a los hijos y les asigna el rol a cada uno
3. Indica las posiciones iniciales
4. Recibe los movimientos de los procesos hijos y se los comunica al otro.

Por lo general, en una topología cliente/servidor como esta, también sería el padre el encargado de validar que los movimientos que pasa el hijo son correctos. En este caso, no será necesario implementarlo.

Ítem 4: Proceso hijo/ratón

Es el que tira primero.

1. Recibe del padre la posición de las piezas.
2. Pinta el escenario (con SFML)
3. Permite realizar la tirada y controla que sea correcta (en un entorno cliente/servidor real, esto último también se haría en el padre)
4. Envía la tirada al padre.
5. Queda a la espera de que el padre le marque el turno y lo que ha tirado el ratón.

Ítem 5: Proceso hijo/gato

Tira en segundo lugar.

Las funciones son las mismas que en el hijo/ratón.

Ítem 6: Recursos necesarios

Para la programación de la estructura de procesos y su comunicación

En este punto te puedes valer de toda la teoría y ejercicios hechos en clase sobre programación de procesos, signals y pipes.

Para la programación de la interfaz de usuario

El objetivo de la asignatura no es la programación de interfaces de usuario. Así que para esta parte está disponible una implementación básica del pintado del tablero, de las piezas y de la captura del evento del mouse.

Puedes encontrar un enlace en el Moddle con un ejemplo de código que puedes abrir directamente con el Code::Blocks. Ya está todo configurado y no es necesario configurar librerías. Se podría empezar a implementar a partir de aquí.

Ítem 7: Consejos

1. No empieces a programar sin tener claro lo que tienes que hacer. Hazte pequeños ejemplos de código donde vayas probando lo que quieres hacer poco a poco. Así te aseguras de que lo que vas incorporando funciona. Por ejemplo, codifica primero la comunicación del proceso padre con un hijo y, luego incorpórale el segundo hijo.
2. El ejemplo de código de la interfaz gráfica es sólo un main.cpp porque no se necesita más. Para tu práctica, organiza el código cómo consideres oportuno: Genera más clases o divídelo en varios proyectos (recuerda que desde el padre puedes llamar a procesos hijos con el `exec/` hacia otra aplicación). Las dos opciones son correctas.
3. No esperes hasta el último día. Seguramente la profe no podrá echarte una mano el día de antes de la entrega.

Ítem 8: Qué hay que entregar

El código

Debe cumplir con los siguientes requisitos:

1. Debe compilar (si no compila, no se corrige y se da por no entregado).
2. Debe permitir el movimiento del ratón y debe comunicarlo al proceso del gato
3. Debe permitir el movimiento del gato y debe comunicarlo al proceso del ratón
4. Debe distinguir a qué jugador le toca tirar. Sólo se permite mover pieza al que tiene el turno asignado.
5. Punto excelencia (para llegar a 10): Se debe detectar e informar del final de partida y del ganador.

Haz un `zip` del proyecto o de los proyectos que hayas utilizado para implementar la práctica.

Procura que el código esté lo más comentado posible.

Da las indicaciones que creas necesarias para ponerlo en marcha.

Memoria (por favor, muy breve)

1. Indica qué procesos se utilizan en tu práctica y cuál es la jerarquía.
2. Indica cómo queda la tabla de *file descriptors* de cada uno de los procesos y para qué se utiliza cada uno.
3. Explica en qué momentos se envía la información y dónde se recibe. Relaciónalo con la zona del código dónde ocurra (indicar la función o la línea de código).
4. Explica si has utilizado *signals* y para qué.
5. Explica si has utilizado *threading* y para qué.
6. Conclusiones: Qué has aprendido con esta práctica, qué te ha costado más, cualquier opinión que consideres relevante...