



Sistemas Operativos

Tarea 1:

Profesor: Viktor Tapia

Ayudante Cátedra: Sebastián Alvarado

Ayudante Tareas: Giorgio Pellizzari

5 de Septiembre, 2019

1 Reglas Generales

Para la siguiente tarea se debe realizar un código programado en lenguaje C. Se exigirá que el código se presente de la forma más limpia y legible posible. Deberá incluir un archivo README con las instrucciones de uso de su programa junto a cualquier indicación que sea necesaria y el makefile del código.

2 Tarea

En esta ocasión se le solicita programar el famoso juego de cartas UNO™, el cual constará de 4 jugadores. Para esto su programa deberá ser capaz de crear 3 procesos hijos (*fork*) los cuales cumplirán con el rol de jugadores (los 3 hijos más el padre). Las jugadas y cambios de turnos deberán ser comunicadas entre jugadores mediante el uso de pipes y deben interactuar con el mazo de cartas, en el cual cada carta sera representada como un archivo .txt.

3 Reglas

3.1 Objetivo del juego

El objetivo de UNO es deshacerse de todas las cartas que se “roban” inicialmente, diciendo la palabra “UNO” cuando queda la última carta en la mano. El primero que elimina todas las cartas de su mano, gana.

3.2 La preparación del juego

El mazo consta de 108 cartas que se deben barajar. Se deben repartir 7 a cada jugador. La primera carta se revela y se dispone como la zona de juego.



3.3 Desarrollo del juego

El primer jugador pone una carta de su mano. Una carta sólo se puede superponer en una carta del mismo color o del mismo tipo. Si se pone una carta que no cumpla esto, se debe robar una carta. Las cartas negras se pueden superponer sobre cualquier carta. Si un jugador no puede poner la carta oportuna, tiene que tomar una carta del mazo. Puede jugar esta carta ahora mismo si existe la posibilidad. Si no, pasa a ser el turno del siguiente jugador. Quién pone su penúltima carta, debe decir “UNO” y señalar que tiene sólo una última carta en la mano. El ganador de la ronda es quien pone su última carta. El juego termina cuando se acaba el mazo o cuando existe un ganador.

3.4 Cartas

Deberá generar las 108 cartas como archivos .txt siendo el nombre del archivo el tipo de la carta (ejemplo: 3 rojo, +4, +2 verde, etc). Estas inicialmente estarán todas en la carpeta “mazo” ordenadas de forma aleatoria. Cada jugador poseerá su propia carpeta de cartas la cual será su mano. Cuando se procede a repartir las cartas por manos, estas deben salir de la carpeta “mazo” y pasar a la carpeta del jugador correspondiente. Además existirá una carpeta donde se dejará solo la última carta revelada, por lo que si un jugador deja una carta ahí, la carta anterior debe ser eliminada. Toda la información del juego debe ser presentada por turnos en la consola, para acceder a la mano del jugador que debe jugar, el programa deberá navegar a la carpeta de la mano del jugador y mostrar su contenido, dándole la oportunidad al jugador de decidir que hacer con sus cartas. El jugador debe poder ver en su turno la última carta revelada, lo cual el programa mostrará accediendo a la carpeta donde se encuentra esta carta y mostrando su contenido.

Los colores existentes son: azul, rojo, verde, amarillo.

- Cartas 1-9: 2 de cada color.
- Cartas 0: 1 de cada color.
- Cartas +2: 2 de cada color.
- Cartas de reversa: 2 de cada color.
- Cartas de salto: 2 de cada color.
- Cartas de colores: 4 total de color negro.
- Cartas +4: 4 en total de color negro.

3.5 Cartas Especiales

- Carta +2: Cuando se pone esta carta, el siguiente jugador debe tomar dos cartas y no puede jugar en su turno (se salta). Esta carta sólo se puede superponer en una carta con el color apropiado u otras cartas “+2”. Si se revela al principio del juego, las mismas reglas aplican.
- Carta de reversa: Con esta carta se cambia la dirección del juego. Es decir que el próximo turno será de la persona que se encuentre en lado contrario y el juego continuará en ese sentido (como cambiar la dirección de las agujas del reloj). La carta sólo se puede superponer en una carta con color correspondiente o en otra carta de reversa. Cuando esta carta se toma en el principio del juego, el primer jugador comienza y entonces el jugador de la dirección contraria continúa.
- Carta de salto: Después de poner esta carta, el siguiente jugador no podrá jugar en su turno (lo salta). La carta sólo se puede superponer en una carta con color correspondiente o en otra carta de salto. Cuando esta carta se toma en el principio del juego, el jugador se “salta” su turno y le toca al próximo jugador.



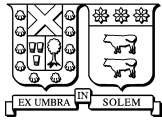
- Carta (negra) de colores: Con esta carta el jugador decide qué color sigue en el juego (este no puede ser negro). Una carta de elección de colores se puede poner sobre cualquier carta y se puede jugar cualquier carta sobre esta (que cumpla con el color elegido). Si se toma una carta de elección de colores en el principio del juego, el jugador deberá elegir un color.
- Carta (negra) +4: El jugador decide qué color sigue en el juego (este no puede ser negro). Además, el siguiente jugador debe robar cuatro cartas y no puede jugar en su turno (se salta). Una carta de +4 se puede poner sobre cualquier carta y se puede jugar cualquier carta sobre esta (que cumpla con el color elegido). Si se toma una carta de +4 en el principio del juego, las mismas reglas aplican.

3.6 Consideraciones

- Los turnos de los jugadores deben comenzar en el orden $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 1 \rightarrow \dots$.
- Cuando a un jugador le quede una carta en la mano, los demás jugadores deberán tener conocimiento de esto. Para ello, a cada jugador le deberá aparecer un aviso declarando a cual jugador o jugadores les queda sólo una carta.
- Las jugadas deben ser comunicadas a los jugadores mediante el uso de pipes. Pueden hacerlo del método que prefieran mientras cumplan esta regla.
- Toda regla que usted conozca del juego y que no aparezcan en este documento deben ser **ignoradas**.
- La información a mostrar puede entregarla en cualquier formato mientras se entienda correctamente y se puedan realizar las jugadas.
- Frente a errores a la hora de ingresar jugadas, el programa debe ser capaz de resolicitar la información.

4 Consideraciones y método de trabajo

- Se deberá trabajar en parejas. Se deberá entregar en Moodle a mas tardar el día 27 de Septiembre del 2019 a las 23:55 horas. Se descontarán 5 puntos por cada hora o fracción de atraso. Las copias serán evaluadas con nota 0 en el promedio de las tareas.
- Aparte de la entrega final se solicitaran dos avances previos, para lo cual se le solicitará realizar la tarea en un repositorio de *github*. Estos avances tendrán el mismo peso ponderado que la entrega final, siendo el promedio geométrico de las tres notas su nota final.
- La tarea debe ser hecha en el lenguaje C. Se asume que usted sabe programar en este lenguaje.
- Los archivos deberán ser comprimidos y enviados juntos en un archivo .tar.gz en el formato TAREA1_ROL1.ROL2.
- Las preguntas deben ser hechas por *Moodle*
- Si no se entrega Makefile o README, o si su programa no compila, la **nota es 0** hasta la corrección.
- Se **descontarán** 50 puntos por:
 - Mala redacción en el README.
 - No funcionamiento de Makefile.
 - No respetar el formato de entrega.



5 Avances

En cada avance deberán presentar código funcional (con makefile), con alguno de los puntos que requiere la tarea para su entrega final. Ustedes deberán determinar que punto de la tarea les conviene abarcar antes que otros. Deberán generar un readme, tanto con dudas como con una explicación de lo que lograron programar.

Los puntos que se recomiendan para avanzar en cada entrega son los siguientes:

- Manejo de archivos (navegación, creación y movimiento de archivos por carpetas).
- Generación de la lógica del juego (implementación de reglas y condiciones de termino).
- Coordinación de procesos (manejo de turnos a través de pipes).

Si su avance hace menos de lo que abarca un punto su nota se verá perjudicada. Ya para el segundo avance, el segundo punto que incorporen debe funcionar en conjunto a lo que habían hecho anteriormente. Si determinan otro punto a avanzar distinto a los recomendados deberán justificarlo apropiadamente en el readme.

La entrega final por moodle evaluará la tarea en su total funcionamiento, sin importar lo que se haya hecho en cada avance.

No se evaluara el uso apropiado de *git* pero si que ambos integrantes colaboren en el repositorio.

5.1 Fechas de avances

- El primer avance se revisará el día 13 de Septiembre (vale decir, se revisará hasta el ultimo commit de ese día)
- El segundo avance se revisará el día 20 de Septiembre.
- Deberán enviar el link del repositorio al siguiente correo: giorgio.pellizzar.14@sansano.usm.cl antes de la fecha del primer avance con nombre de asunto: TAREA1-SO-2019-2