Instituto de Tecnología ORT

Carrera: Analista de Sistemas

Materia: Fundamentos de Programación

EXAMEN FINAL 1 **Tema 2** **02/12/2021**

😃 **Leé por lo menos dos veces el enunciado antes de resolverlo.**

Enunciado

“Luz verde, luz roja”

Una organización clandestina lleva a cabo una competencia donde los jugadores ponen en riesgo su vida a cambio de una importante suma de dinero para el único ganador/superviviente.

Cada jugador se identifica con un número entero unívoco. De hacer falta podrán agregarse todos aquellos atributos que se consideren necesarios para resolver el comportamiento descripto a continuación.

Nos piden simular uno de estos juegos mortales. Este juego se conoce como “Luz verde, luz roja”.

Este juego consiste en cruzar de una punta a la otra una distancia de **150 metros**. Todos los jugadores están en un extremo (el metro cero). En el extremo opuesto de la pista (allí donde los jugadores deben llegar) se encuentra una muñeca gigante. Cuando la muñeca se encuentra de espaldas a los jugadores hay “luz verde” y se puede correr hacia la meta y la salvación. Pero cuando la muñeca se da vuelta para enfrentar a los jugadores se decreta la “luz roja” y todos los jugadores deben detenerse y quedarse “congelados”. Todo aquel jugador que se detecte en movimiento será eliminado.
En total, la muñeca completará **10 rondas** cambiando de estado, uno de cada tipo por ronda: primero “luz verde”, cuando los jugadores serán libres de moverse y avanzar todos los metros que puedan. Cuando la muñeca gire para enfrentarlos se prenderá la luz roja. Entonces, la muñeca los "observará", revisando uno a uno a los jugadores para detectar quiénes están en movimiento y entonces eliminarlos.
Todo jugador que alcance la meta estará a salvo y deberá moverse a una zona de jugadores supervivientes. También se apartarán (a otro lado) aquellos que sean eliminados. Al cumplirse la décima luz verde y prenderse la luz roja todos aquellos jugadores que no hayan conseguido alcanzar la meta también serán eliminados.

Métodos provistos:

Para saber cuántos metros avanzó un jugador en una ronda se provee el método:
`private double obtenerAvance()` de la clase Jugador, el cual no recibe parámetros y siempre devuelve un valor que indica el avance del jugador durante una luz verde.

Para saber si un jugador está quieto en una ronda, se provee el método:
`public boolean estaQuieto()` de la clase Jugador, el cual no recibe parámetros y devuelve un valor booleano indicando si el jugador está quieto o no.

Estos dos métodos provistos NO deben diagramarse en NS+, solamente deben agregarse al diseño e invocarse en donde se considere necesario al desarrollar los métodos requeridos en NS+.

Instituto de Tecnología ORT

Carrera: Analista de Sistemas

Materia: Fundamentos de Programación

EXAMEN FINAL 1 **Tema 2** **02/12/2021**

Se pide:

- Confeccionar el diagrama UML que describe el escenario del enunciado incluyendo los atributos de cada clase y los métodos a desarrollar y aquellos que creas conveniente.
- Desarrollar el método **`competir(...)`** de la clase **Juego** que recibe como parámetro dos colecciones provisorias de jugadores, ambas instanciadas pero vacías y que deberán ser cargadas durante el juego de la siguiente manera:
  - En la primera colección deben quedar cargados aquellos jugadores que sobrevivieron.
  - En la segunda colección deben quedar cargados los jugadores que fueron eliminados.

Nota 1: *Para desarrollar el juego, de cada jugador también se deben saber el número de ronda en el que llegó a la meta ó en la que fue eliminado, y la cantidad de metros recorridos.*
Nota 2: *En el desarrollo de este método se dará especial énfasis a la correcta modularización y distribución de tareas y/o responsabilidades.*

Importante: El juego posee una colección con todos los jugadores que participarán del mismo. Al finalizar la ejecución del método ***"competir"*** dicha colección debe quedar vacía, dado que los jugadores debieron haber sido movidos a alguna de las dos colecciones resultantes.

Criterios

Para considerar aprobado el examen, el mismo debe resolver lo pedido y aplicar los siguientes conceptos de la programación orientada a objetos:

- Detección de clases, atributos, métodos y relaciones (asociativas y de uso).
- Modularización reutilizable y mantenible usando métodos con correcta parametrización.
- Asignación de responsabilidades a cada clase y correcto encapsulamiento.
- Manejo del concepto de instancia y de la interacción entre objetos.
- Manipulación de listas de objetos (ArrayList) y su uso en ciclos condicionales y for-each, eligiendo siempre el más adecuado en cada situación.
- Manejo de diagramas Nassi-Schneiderman y UML de clases.

% Correcto	0 a 20	25 a 45	50 a 55	60	65 a 70	75	80	85 a 90	95	100
Nota	1	2	¿4?	4	5	6	7	8	9	10