Proyecto I – Math socket

Instituto Tecnológico de Costa Rica
Área de Ingeniería en Computadores
Algoritmos y Estructuras de Datos I (CE 1103)
II - 2021
Valor 25%

Objetivo General

- ➔ Diseñar e implementar un juego de mesa multijugador.

Objetivos Específicos

- ➔ Implementar pilas, colas, listas enlazadas y algunas variaciones.
- ➔ Investigar y desarrollar una aplicación en el lenguaje de programación Java.
- ➔ Investigar acerca de programación orientada a objetos en Java.
- ➔ Aplicar **patrones de diseño** en la solución de un problema.
- ➔ Utilizar UML para modelar la solución de un problema.

Descripción del problema

Math socket es un juego de mesa para dos jugadores simultáneos. El juego consiste en un tablero de *nxn,* donde el tamaño *n* es definido por los estudiantes, pero debe ser mínimo de 4x4 y máximo de 6x6. Dentro de dicho tablero existirán tres tipos diferentes de casillas:

1. Casilla de reto
2. Casilla de túnel
3. Casilla de trampa

Cada una de las casillas tendrá una función particular que será definida más adelante. La figura 1 muestra un ejemplo del tablero de juego.

Debido a que el flujo del juego no es siempre hacia adelante, el tablero deberá ser representado como una lista doblemente enlazada, donde la información del nodo corresponderá al tipo de casilla que representa y su función. Por ejemplo, una casilla de reto debe almacenar el reto que se le planteará al otro usuario. La definición de casillas será aleatoria, pero se debe procurar mantener una proporcionalidad adecuada entre casillas de reto y casillas de trampa y túnel. Se espera que al menos el 50% de las casillas sean de reto. El otro 50% deberán ser casillas de trampa y túnel. El estudiante debe definir de antemano el tamaño de su tablero y la cantidad de casillas de reto y casillas de trampa y túnel, pero la colocación de las casillas en el tablero será aleatoria.

La comunicación entre los dos jugadores deberá ser a través de sockets, donde un jugador será el servidor (el jugador que inicia la partida) y el otro el cliente (el jugador que se une a la partida).

Este es un juego de turnos, donde en cada turno el jugador puede “tirar un dado” y este le dará el número de casillas que puede avanzar. Dado que la cantidad de casillas no es tan grande, el dado no podrá dar

números mayores a 4. Ganará el jugador que llegue primero al final del tablero.

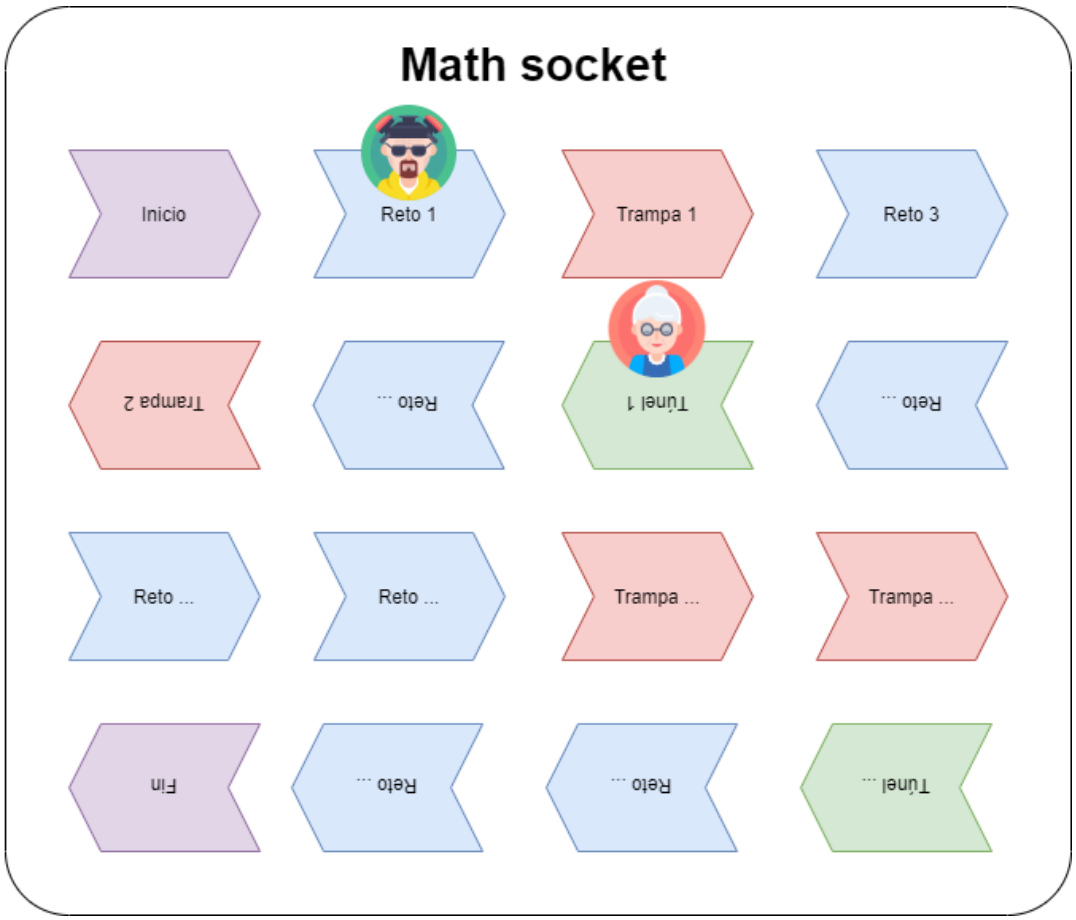

Figura 1. Ejemplo de tablero de juego.

Descripción de casillas

Casilla de reto

Este tipo de casillas son las más comunes en el tablero, y la idea es poder ponerle un reto matemático al otro jugador. Si el otro jugador falla la respuesta, él se devuelve una casilla; pero si acierta la respuesta, mantendrá su posición actual.

En cualquier caso (si el otro jugador acierta o no), el jugador en cuestión avanza un paso hacia adelante.

Los retos consisten en resolver operaciones matemáticas algebráicas simples, que involucren sumas, restas, multiplicación y división **de únicamente dos operandos enteros**. Los retos serán de la forma

$$x \otimes y$$

donde x y y son los operandos enteros y $\otimes$ corresponde al operador (suma, resta, multiplicación o división)
Los dos operandos deben ser aleatorios en un rango de 1 hasta 50.

Casilla de túnel

Esta casilla permite avanzar al jugador una determinada cantidad de casillas hacia adelante. La cantidad de casillas a avanzar deberá ser aleatoria, pero estará en el rango de 1 a 3. Esto significa que no se puede avanzar menos de una casilla y más de tres.

Casilla de trampa

Esta casilla funciona igual que la casilla de túnel, pero en lugar de avanzar hacia adelante, se mueve hacia atrás.

Descripción de ventanas

Ventana de inicio

Esta ventana funcionará únicamente para esperar a que todos los jugadores están conectados a la partida. Aquí se podrá escoger un nombre para el jugador.

Ventana de juego

Esta ventana desarrollará el juego en cuestión, y finalizará cuando alguno de los dos jugadores gane la partida. Cuando el jugador contrario presente un reto, la ventana de juego deberá mostrar el reto en cuestión y deberá tener un espacio para ingresar la respuesta.

El otro jugador deberá validar que la respuesta sea correcta automáticamente.

Aspectos operativos y evaluación

1. **Fecha de entrega: De acuerdo al cronograma del curso**
2. El proyecto tiene un valor de 25% de la nota del curso.
3. El trabajo es **en grupos de 3 personas**.
4. Es obligatorio utilizar GitHUB.
5. Es obligatorio integrar toda la solución.
6. El código tendrá un valor total de 85% y la documentación 15%.
7. De las notas mencionadas en el punto anterior se calculará la Nota Final del Proyecto.
8. Se evaluará que la documentación sea coherente, acorde a la dificultad/tamaño del proyecto y el trabajo realizado, se recomienda que realicen la documentación conforme se implementa el código.
9. La nota del código NO podrá exceder en 35 puntos la nota de la documentación, por lo cual se recomienda documentar conforme se programa.
10. La documentación se revisará según el día de entrega en el cronograma.
11. Las citas de revisión oficiales serán determinadas por el profesor durante las lecciones o mediante algún medio electrónico.

12. Los estudiantes pueden seguir trabajando en el código hasta 15 minutos antes de la cita revisión oficial
13. Aún cuando el código y la documentación tienen sus notas por separado, se aplican las siguientes restricciones
    a. Si no se entrega documentación, automáticamente se obtiene una nota de 0.
    b. Si no se utiliza un manejador de código se obtiene una nota de 0.
    c. Si la documentación no se entregan en la fecha indicada se obtiene una nota de 0.
    d. Sí el código no compila se obtendrá una nota de 0, por lo cual se recomienda realizar la defensa con un código funcional.
    e. El código debe ser desarrollado en Java, en caso contrario se obtendrá una nota de 0.
    f. Si no se siguen las reglas del formato de email se obtendrá una nota de 0.
    g. La nota de la documentación debe ser acorde a la completitud del proyecto.
14. La revisión de la documentación será realizada por parte del profesor, no durante la defensa del proyecto. El único requerimiento que se consultará durante la defensa del proyecto es el diagrama de clases, documentación interna y la documentación en el manejador de código.
15. Cada grupo tendrá como máximo 15 minutos para exponer su trabajo al profesor y realizar la defensa de éste, es responsabilidad de los estudiantes mostrar todo el trabajo realizado, por lo cual se recomienda tener todo listo antes de ingresar a la defensa.
16. Cada excepción o error que salga durante la ejecución del proyecto y que se considere debió haber sido contemplada durante el desarrollo del proyecto, se castigará con 2 puntos de la nota final del proyecto.
17. Durante la revisión únicamente podrán participar el estudiante, asistentes, otros profesores y el coordinador del área.