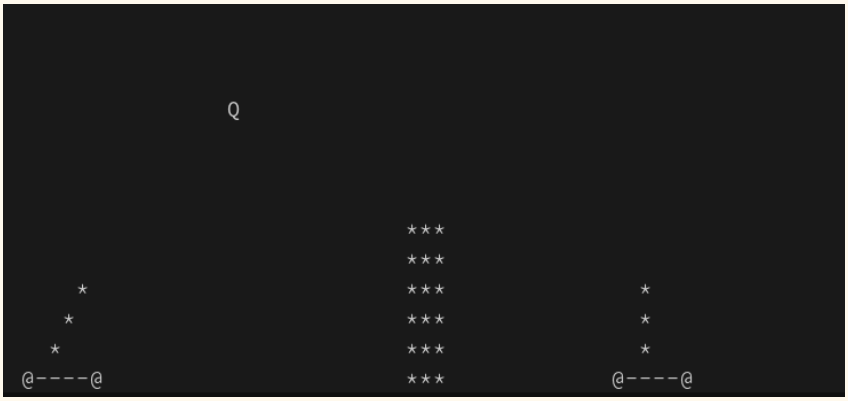
2024, INFO188 Tarea 1: CANOWARS

Profesor: Cristóbal A. Navarro**, Ayudante:** Alejandro Villagrán

Enunciado

En esta tarea, usted implementará en Haskell un videojuego de combate "versus" que se puede jugar entre dos personas en un mismo teclado (juegos antiguos se jugaban así), llamado **CANOWARS**, jugado por el terminal. El juego consiste en una batalla entre dos catapultas, donde el ganador es el que logra destruir al rival con proyectiles. Requisitos:

1. El escenario es en 2D (visto desde lado, no desde arriba), dividido por una muralla al medio. Es decir, una mitad es el territorio de una catapulta, la otra mitad es de la otra catapulta.
    a. REFERENCIA (use su propio estilo, quizás quieran hacer el escenario más grande.)



2. Cada catapulta debe ser dibujada con los caracteres ascii. Diseñe algo creativo.
3. El juego es por turnos en un mismo teclado. Decida usted cuales son las teclas del player 1, y 2.
4. Cada catapulta puede orientar el ángulo de su lanzamiento (averigue cómo lograrlo).
5. Cada catapulta tiene 30 HP, el proyectil hace entre 1-3 de daño. Hay 5% de crítico (7-9 de daño).
6. Cada catapulta tiene una cantidad finita de combustible para usar por turno:
    a. 100 de combustible por turno.
    b. Desplazarse, mover el cañón, y disparar, usan combustible. Decida los gastos.
7. Al disparar, el proyectil de la catapulta debe seguir y mostrar una trayectoria parabólica con ángulo de salida **β** + RAND(-5,+5), donde **β** es el ángulo del canon en ese momento. Manejar situaciones de borde.
8. El videojuego debe ejecutarse por terminal así: `./canowars`.
9. Si encuentra limitaciones en el camino entonces explíquelas en su informe.
10. **NOTA:** Hay muchos puntos en donde deben tomar decisiones, por ende cada grupo debería terminar con una versión distinta del videojuego.

IMPORTANTE:

1. Pueden usar asistentes de IA como chat GPT para acelerar el desarrollo.
2. Su videojuego debe ser un programa con **main,** un **módulo *.hs** y un **Makefile**.
3. Incluir manual de compilación, ejecución, librerías necesarias etc.
4. Debe usar **al menos un tipo de dato** propio.
5. Se evaluará el **buen uso del paradigma funcional**.
6. **Documente** todo su código con explicaciones arriba de cada función, y comentarios entre líneas en las partes sean más difíciles de seguir.
7. Debe incluir **al menos una abstracción funcional**; Functor, Aplicativo, o Monada.
8. **[BONUS NOTA]** más abstracciones funcionales usadas, gameplay mejorado (sonido color etc).

Grupos, entregables y fecha de entrega

- **[GRUPOS]** Pueden trabajar en grupos de hasta 4 personas.
- **[ENTREGABLES]** Archivo ***.zip** con código fuente + informe PDF de máximo 4 páginas.
- **[PLAZO]** **10 de Noviembre 2024, todo el día.**