

Ahorcado ft Genserver

Fecha de entrega: 8/Septiembre

Introducción

Para esta actividad vas a completar el juego del ahorcado visto anteriormente, pero con unas funcionalidades adicionales e integrando Genserver para que este sea de ayuda cuando el jugador requiera una pista para seguir adivinando la palabra en juego.

Tendrás a disposición el juego avanzado en un 60% y te corresponde terminar el 40% del mismo que corresponde a la integración con GenServer.

Son 2 archivos con los cuales se debe trabajar: `hangman_server.ex`, `hangman_utils.ex`.

Para trabajar con estos archivos, primero debe compilarse `hangman_utils.ex` y luego `hangman_server.ex`

Contexto

Como es de su conocimiento, el ahorcado trata de ir adivinando letra por letra hasta dar con la palabra completa. Si adivina la letra, ganará puntos de acuerdo a si es una vocal o una consonante acorde a la cantidad de veces que se encuentre en la palabra.

- Vocal: 5 puntos.
- Consonante: 3 puntos.

Adicional, el usuario podrá solicitar una pista hasta 3 veces, sin embargo, esto no debe ser sumado a los puntos actuales del jugador ni contado como oportunidad. Evidentemente, el dibujo tampoco se altera. Esto es controlado por medio de GenServer.

Por hacer

memory_utils.ex

start_server()

Iniciar el servidor - 1p

init(_param)

Definir el estado inicial de 3 para ir controlando la solicitud de pistas - 1p

handle_call({word_game, selW}, _from, status)

Desarrollar la lógica para ir disminuyendo la cantidad de pistas y mostrar un mensaje cuando queden 'n-pistas' por pedir o ninguna. - 5p

NOTA: El proceso de pedir la pista y actualizar la palabra que está jugando ya está realizado y almacenado en la variable word_up.

give_clue(word_game, selW)

Realizar el llamado correcto para llamar a handle_call - 1p

game(word_game, selW, oport, ptos, drawing)

En esta función está la lógica del juego al 90%, para completarla debes realizar lo siguiente

- En la estructura case de la línea 24, Incluir el caso para solicitar pistas al servidor (llamar a give_clue) y que calce con la lógica del case para seguir llamando a game() - 1p
- En las definiciones de terminación de la función, editar lo que realizan de modo que se detenga el servidor llamando a stop_server(result) y mostrando el resultado final de la partida - 1p

Restricciones e indicaciones generales

1. Lo entregado está 100% operativo y lo que se solicita es para completar la funcionalidad del juego.
2. Puede crear funciones adicionales de ser necesario.
3. El uso de if está limitado a una única vez, si lo utiliza más veces será penalizado con reducción de puntos.
4. El uso de for está prohibido.
5. El consolidar los dos archivos en un único archivo equivale a una nota de cero puntos.
6. Considere los HINTS dados y no dude en consultar en la documentación.
7. Si edita lo entregado por el coach corre el riesgo de corromper la funcionalidad, a menos que sepa arreglarla o mejorarla.

Entregables

- Video de no más de 5 minutos que debe incluir (En este orden):
 - El juego funcionando 100%
 - Explicar la integración con GenServer.
- Enlace a repositorio github con los 2 archivos .ex.