Tecnológico de Costa Rica

Escuela de Computación

IC-4700 Lenguajes de programación

Documentación - Proyecto #1:

Programación Imperativa

Estudiantes:

Jorge Arturo Chavarría Villarevia- 2023157488

Murillo Chaves Felipe Javier - 2023083005

Hidalgo Sánchez Fernando Andrés - 2022218688

Grupo 02

Profesor:

Bryan Tomas Hernandez Sibaja

I Semestre

2025

Índice

Enlace a GitHub

https://github.com/ArturoChV10/proyectoLenguajes.git

Descripción

El proyecto #3 para el curso de Lenguajes de programación tiene como objetivo familiarizar a los estudiantes con el desarrollo de programas utilizando el paradigma lógico, para esto se solicitó el desarrollo de un juego de “Ahorcado” utilizando Prolog. Una vez se inicializa el juego se ejecutarán tres estados principales:

- Menú principal: Inicialmente, el programa presentará un menú principal, donde el usuario tendrá la posibilidad de configurar ciertos aspectos específicos relacionados a la jugabilidad.
- Mecánica de juego: El programa presenta un sistema de turnos para adivinar una letra de la palabra oculta, misma que presentará las letras desconocidas utilizando guiones bajos, adicionalmente, en este estado, se realizan validaciones de letras repetidas y se tendrá un contador para la cantidad de intentos.
- Finalización: Para finalizar el juego existen dos posibles resultados, el primero corresponde a adivinar todas las letras de la palabra, lo cual genera un resultado positivo, por otro lado, el otro resultado posible corresponde a agotar todos los intentos definidos por el usuario.

Pasos de instalación y ejecución

A continuación, se presentarán diversos pasos para la instalación y ejecución del programa sin problemas mayores, para esta sección, asumiremos un paso 0, y este corresponde a la instalación previa de SWI-Prolog, de modo que el programa tendrá un editor sobre el cual poder ser ejecutado, una vez hemos definido la instalación de Prolog, el paso 1 corresponde a clonar el repositorio de github, ya que mediante el comando:

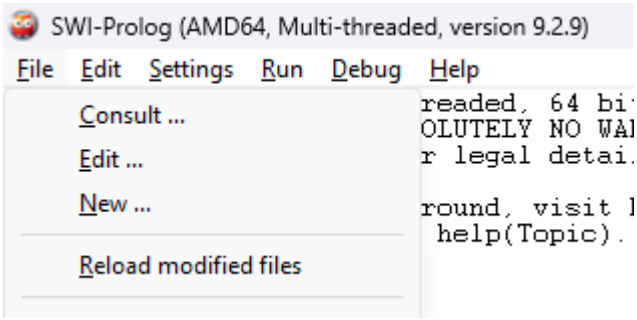
“git clone https://github.com/ArturoChV10/proyectoLenguajes.git”

Podremos obtener todos los documentos necesarios para la ejecución de nuestro programa. El segundo paso en la ejecución del programa corresponde a ejecutar SWI-Prolog y realizar una consulta sobre el documento denominado “muneco.pl”, ya que este es el documento que almacena toda la implementación necesaria del programa, finalmente, el tercer paso corresponde a (una vez realizada la consulta) ejecutar el comando “jugar.”, el cual inicia la ejecución de todo el programa.

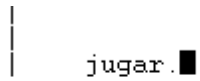
Manual de Usuario

En esta sección se presenta el manual de usuario para la aplicación desarrollada de “Ahorcado”. Aquí se explica brevemente la función del sistema y sus componentes, con el objetivo de facilitar el uso de la aplicación. Este manual servirá como guía de referencia para facilitar su uso. Aunque el documento cumple con un aspecto formal, se ha redactado para la comodidad del usuario final.

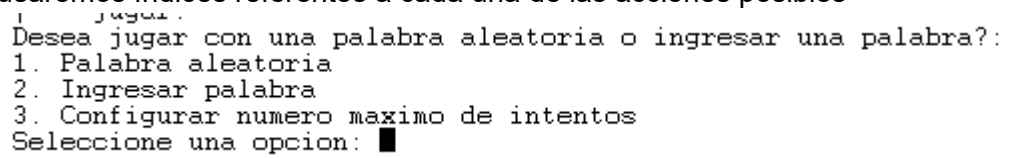
Una vez se hayan descargado previamente todos los componentes necesarios, el primer paso corresponde a hacer la consulta y verificar que el documento “muneco.pl” existan



Una vez abierta esta consulta y que la misma se haya realizado correctamente un mensaje similar al siguiente debería ser desplegado en la pantalla:
d:/chava/Descargas/proyectoLenguajes/ParadigmaLogico/muneco.pl compiled 0.00 sec, 26 clauses
Realizados los pasos anteriores, el siguiente paso corresponde a utilizar la instrucción “jugar.” para poder ejecutar el programa.



El siguiente paso corresponde a seleccionar una de las opciones del menú principal, para esto usaremos índices referentes a cada una de las acciones posibles



Si se escoge la función 1, el programa seleccionará, de manera aleatoria, una de las palabras definidas implícitamente dentro del programa, por otro lado, la función 2 nos permite ingresar una palabra a nuestra escogencia, siendo sobre esta la que se realizará el minijuego, finalmente,

la última opción (3) nos permite modificar el número máximo de intentos (si no es especificado, el valor por default corresponde a 7).

Una vez escogida una de las funciones 1 o 2, el sistema presentará una interfaz similar a la siguiente:

```
Comienza el juego!
Intentos restantes: 7
Letras usadas: []
Escoja una letra:
|:
```

Aquí, los usuarios tomarán turnos para adivinar una letra por turno, avanzando en el programa cuantos más intentos se realicen.

Si al ingresar una letra, esta no se encuentra en la palabra, un mensaje como el resaltado en rojo será mostrado, mientras que si la letra sí se encuentra, el sistema enviará un mensaje como el resaltado en verde:

```
Seleccione una opcion: 1
Comienza el juego!
Intentos restantes: 7
Letras usadas: []
Escoja una letra:
|: a
La letra a NO se encuentra en la palabra
=========
|
|
|
|
|
|
=========
_ _ _ _ _
Intentos restantes: 6
Letras usadas: [a]
Escoja una letra:
|: e
La letra e se encuentra en la palabra
_ _ _ _ e
Intentos restantes: 6
Letras usadas: [a,e]
Escoja una letra:
|:
```

Finalmente, si se gana o se pierde el minijuego de “Ahorcado” un mensaje de felicitaciones o de consuelo será desplegado en la consola, indicando el fin de la ejecución

.

Explicación del funcionamiento

El proyecto fue desarrollado bajo el paradigma lógico, utilizando Prolog como lenguaje de programación principal. En este enfoque, el conocimiento se representa mediante hechos y reglas, y el motor lógico del lenguaje se encarga de realizar inferencias a partir de consultas. La lógica del programa se estructura en base a relaciones entre hechos, lo cual permite describir el comportamiento del juego de manera declarativa.

Estructura General del Programa

El programa está compuesto por una serie de instruccuibes lógicas definidas en el archivo muneco.pl. Este archivo contiene:

1. Hechos base:
   - Palabras predefinidas para el juego (solo cuando se utiliza la opción de palabra aleatoria).
   - Letras válidas del alfabeto.
2. Reglas de juego:
   - Validación de entrada del usuario.
   - Selección de palabras.
   - Reglas para determinar si una letra ha sido adivinada correctamente o no.
   - Actualización del estado del juego según los intentos y letras adivinadas.
3. Predicados principales:
   - jugar/0: Es el punto de entrada del programa. Ejecuta el flujo general del juego.
   - menu/0: Despliega las opciones principales y redirige según la decisión del usuario.
   - seleccionar_palabra/1 y pedir_palabra/1: Manejan la selección automática o manual de palabras.
   - jugar_turnos/4: Controla la lógica de turnos, mostrando el estado actual de la palabra e interpretando las acciones del usuario.
   - verificar_letra/3: Evalúa si la letra ingresada se encuentra en la palabra y actualiza el progreso.
   - estado_juego/3: Determina si se debe continuar el juego, mostrar victoria o derrota.

Flujo de Ejecución

1. Inicio:
   - El usuario ejecuta jugar. en la consola tras haber consultado el archivo.
   - El sistema despliega el menú principal.
2. Configuración inicial:
   - El usuario elige entre usar una palabra aleatoria, ingresar una personalizada o cambiar el número de intentos permitidos.
   - Según la opción, se configura el estado inicial del juego: palabra oculta, lista de letras adivinadas vacía e intentos disponibles.
3. Mecánica del juego:
   - Se inicia un ciclo de turnos con el predicado jugar_turnos.
   - En cada turno:
     - Se muestra el progreso actual de la palabra (letras adivinadas y guiones bajos).
     - El usuario ingresa una letra.
     - Se verifica si ya fue usada o no.
     - Se actualiza el estado del juego según si la letra fue correcta o incorrecta.
     - Se actualiza el conteo de intentos si es necesario.
4. Finalización:
   - Si todas las letras de la palabra han sido adivinadas, se muestra un mensaje de victoria.

- Si se agotan los intentos, se despliega un mensaje de derrota.
- En ambos casos, se termina la ejecución del juego.