

Proyecto Programado

Programación Lógica en Prolog.

1. Motivación

Como actividad para la puesta en práctica de programación lógica, mediante el lenguaje Prolog, se propone el desarrollo de un juego de acertijos para niños.

2. Objetivos Formativos.

La presente asignación pretende servir como herramienta para que las y los estudiantes logren el objetivo de aprender Prolog como lenguaje ejemplo para el paradigma de programación lógica. Además de estudiar de forma práctica cómo se pueden crear ligas entre soluciones en paradigmas y lenguajes de programación distintos para conformar un solo producto final.

3. Metodología.

En grupos de trabajo de tres estudiantes, las y los estudiantes deberán modelar, diseñar, programar y documentar su propia versión del juego de mesa OutFoxed¹. La lógica del juego deberá ser implementada utilizando Prolog mientras que la interacción con el usuario debe programarse generando una interfaz gráfica con Java. Outfoxed es un juego para niños² en el que un personaje comete un robo y los jugadores deben ir descartando sospechosos y buscando pistas para encontrar quién es el ladrón antes de que este escape.

El juego cuenta con 16 personajes, todos son iguales excepto por los accesorios y prendas de vestir que usan.

Para efectos de este proyecto, los grupos de trabajo deberán definir:

¹ <https://www.fgbradleys.com/rules/rules4/Outfoxed%20-%20rules.pdf> Reglas del Juego Outfoxed.

² <https://www.youtube.com/watch?v=fATRnRnRwLw> Video Explicativo del Juego Outfoxed.

- Objeto robado.
- Personaje base. (equivalente al zorro).
- Dieciséis combinaciones distintas para cada uno de los personajes. (con 3 accesorios/prendas de vestir distintos).
- Escenario en el que los hechos se llevan a cabo.
- Tablero del juego (ruta de escape y posiciones de las pistas).
- Doce pistas (prendas de vestir o accesorios).
- Decifrador de pistas.

Se recomienda estudiar el funcionamiento del juego original de manera minuciosa para definir el comportamiento del programa a implementarse, tomando en consideración que en este caso será el programa el que busque pistas hasta que pueda indicar quien es el ladrón. Para lo anterior, consideren implementar procedimientos dinámicos³, mediante los cuales se puedan agregar⁴ o eliminar⁵ hechos en tiempo de ejecución, según se requiera.

Se recomienda también trabajar con JavaFX⁶ para manejar la interacción con el usuario y la con biblioteca JPL⁷ para manejar la interfaz que comunica al programa en Java con el programa en Prolog.

4. Especificación.

Cada grupo deberá presentar los siguientes entregables, según las fechas establecidas en la siguiente sección.

- 1. Repositorio remoto:** Se deberá configurar un repositorio grupal en GitHub, de modo que todos los integrantes del grupo puedan clonar el proyecto y hacer sus aportes desde su máquina local. Se deberá incluir además a la profesora bajo el nombre de usuaria **tec-ramijan**. Dicho repositorio contendrá:

- El código del acertijo, archivo escrito en el lenguaje de programación Prolog con extensión .pl.
- Proyecto en Java, con la integración de la interfaz y el acertijo..
- Un archivo **README.md** en el que se incluya una breve descripción del proyecto, pasos necesarios para la ejecución del programa y los enlaces a los videos explicativos de todos los integrantes del grupo.

³ <http://www.cse.unsw.edu.au/~billw/dictionaries/prolog/dynamic.html> Procedimientos dinámicos

⁴ <http://www.cse.unsw.edu.au/~billw/dictionaries/prolog/assert.html> Assert

⁵ <http://www.cse.unsw.edu.au/~billw/dictionaries/prolog/retract.html> Retract

⁶ <https://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-overview.htm#JFXST784> JavaFX

⁷ https://www.swi-prolog.org/pldoc/doc/SWI_/library/jpl.pl Interfaz Java + Prolog

Debido a que el repositorio muestra las contribuciones de cada estudiante, así como la hora y fecha de los cambios realizados, es obligatorio actualizar de forma periódica y no solamente al finalizar el proyecto.

- 2. Video explicativo:** Cada miembro del grupo por separado deberá realizar un video, capturando la pantalla de su computadora. En el video debe explicar el diseño de la solución, debe mostrar y explicar sus aportes de código al proyecto y además hacer una demostración de una ejecución exitosa del código. El video deberá ser de entre 10 y 15 minutos. Cada estudiante deberá realizar su propio video, publicarlo a través de YouTube e incluir la URL del video en el archivo `README.md`
- 3. Documentación:** Cada grupo de trabajo presentará un documento que deberá completarse según el calendario presente en la siguiente sección.
 - a. Portada: Nombre de la institución, la sede, nombre del curso y la profesora, semestre actual, integrantes del grupo (con sus respectivos nombres y carnets).
 - b. Diseño de la solución: En esta sección se deben describir en prosa las partes indicadas a continuación, puede incluirse diagramas de flujo y/o representaciones gráficas que ayuden a explicar la solución.
 - i. Hechos.
 - ii. Inferencias Lógicas.
 - iii. Flujo del programa.
 - c. Diseño de la interfaz gráfica: Bocetos o capturas de pantalla de la interfaz a implementar, con los elementos seleccionados para la variación del juego.
 - d. Lecciones aprendidas: En esta sección se presentan las lecciones organizacionales, técnicas y teóricas que se hayan aprendido durante la elaboración del proyecto.

5. Calendario.

Entregable	Fecha límite de Entrega
Acceso al repositorio remoto para les estudiantes que participan y la profesora.	Miércoles 19 de mayo (7 pm)
Diseño de la Solución Lógica. Diseño de la interfaz gráfica a implementar. Lógica del juego en prolog (70% completado).	Miércoles 26 de mayo (7 pm)
Código completo en el repositorio	Miércoles 2 de junio (7 pm)
Video Explicativo	Jueves 3 de junio (7 pm)

6. Rúbrica.

Criterio	Indicadores de rendimiento		
	Muy bien	Regular	Deficiente
Repositorio	Actualizaciones periódicas del repositorio (5). Colaboración equitativa, todos los miembros del grupo realizan aportes equivalentes. (10). 15		
Documentación	Las explicaciones de la solución son claras y fáciles de entender (5) la documentación y la solución del código se basan una en la otra (5). 10		
Interfaz Gráfica (Java)	La solución implementada incluye una representación gráfica para todas las acciones presentes en el juego original. (20) La interfaz es intuitiva, fácil de usar. (5). La propuesta del juego implementada es creativa. (5) 30		
Programa en Prolog	El estilo de programación sigue las normativas del paradigma lógico. (10). El descifrador lógico en Prolog permite validar las pistas y el ladrón al final. (10) Se emplean procedimientos dinámicos para modificar los hechos en tiempo de ejecución. (15) 35		

Liga entre Java y Prolog	La liga entre ambos lenguajes es transparente para la persona usuaria.	El producto final conecta en gran medida ambos lenguajes, pero son necesarias algunas acciones externas para que funcione.	No se logra conectar a ambos lenguajes.
	10	6	0
Video Explicativo (calificación individual)	El video cumple con todos los lineamientos solicitados en este documento.	El video no cumple con algunos de los lineamientos solicitados en este documento.	El video es poco informativo o evidencia falta de claridad por parte del estudiante.
	10	6	4
Puntaje total	100 puntos	Valor porcentual	15%