

Segundo Proyecto de Programación Declarativa

Carlos Toledo Silva C-311
Ariel A. Triana Pérez C-311

1. Introducción

En el siguiente documento se realizará la presentación de la aplicación Loki Text Adventure. Para esto dividiremos el resto del documento en secciones. En una primera sección estaremos hablando sobre la historia elaborada para la aplicación y en una posterior sección nos referiremos a como un usuario interactúa con la aplicación. Posteriormente nos estaremos refiriendo a la implementación de la aplicación: estrategia que se siguió, descripción de los módulos implementados y como se aprovecharon las características propias de Haskell para dicha implementación.

2. Historia elaborada para la aplicación

La historia que se narra en la aplicación está basada en la mitología nórdica. El usuario asume el papel de Loki: un dios embaucador de esta mitología y uno de los dioses más conocidos. El usuario, metido en el papel de Loki, tomará parte en una historia en la que pasará por diferentes aventuras y dificultades, donde las decisiones que tome y los actos que realicen tendrán una influencia directa en la historia del mundo.

Válido aclarar que aunque la historia está basada en la mitología nórdica, muchos de los pasajes que en ella se narran no ocurren exactamente igual, ni siquiera en el orden correspondiente a como están realmente recogidos en la mitología. Otros muchos ni siquiera ocurrieron y son invención de los autores. De esta forma se ha adaptado la historia de Loki y otros personajes nórdicos para lograr hacer una historia entretenida y atractiva para el usuario.

La historia está compuesta de 3 capítulos, seguidos uno a continuación del otro. Es decir que para el usuario poder llegar el capítulo $i + 1$ deberá haber triunfado en el capítulo i . Además cada capítulo cuenta con varios finales; en algunos el jugador sale triunfante pero en otros le podría acabar perdiendo la vida. Por esto motivo el jugador deberá escoger sabiamente, si es que quiere avanzar lo más posible.

3. Interacción entre el jugador y la aplicación

Cuando se ejecuta la aplicación (ya compilada) lo primero que se ve es lo siguiente:

```
Bienvenido a Loki Text Adventure

Capítulo 1
Eres Loki, Señor del Caos, hijo de los gigantes Farbuti y Laufey. Se te conoce hasta ahora como conflictivo, egocéntrico y megalómano. Tu actuación en esta historia llevará a que reine la paz en el mundo o este se sumerja en un caos total. Tienes tres hijos el enorme lobo Fenrir, la diosa de la muerte Hela y la serpiente marina Jormundgander. Tu historia comienza en Asgard, estas caminando por el medio de la plaza cuando pasas cerca de una casa con la puerta abierta y una luz encendida.
```

Aquí es donde comienza la historia. A medida que el personaje avance en la historia le irán saliendo textos similares a estos. Para poder avanzar el usuario deberá utilizar la información mostrada en cada uno de los textos y dar una respuesta lógica y bien escrita mediante alguna acción. Es importante recalcar que las acciones se deben expresar mediante infinitivos: entrar, aceptar, hablar, etc y que por supuesto estos se combinen con las otras palabras necesarias para elaborar una idea coherente. También es importante recalcar que las acciones sobre el propio jugador deben ir seguidas del pronombre personal me; ejemplo: transformarme, quedarme, etc.

Por ejemplo partiendo del texto inicial podemos hacer lo siguiente:

```
Bienvenido a Loki Text Adventure

Capítulo 1
Eres Loki, Señor del Caos, hijo de los gigantes Farbauti y Laufey. Se te conoce hasta ahora como conflictivo, egocéntrico y megalómano. Tu actuación en esta historia llevará a que reine la paz en el mundo o este se sumerja en un caos total. Tienes tres hijos el enorme lobo Fenrir, la diosa de la muerte Hela y la serpiente marina Jormundgander. Tu historia comienza en Asgard, estas caminando por el medio de la plaza cuando pasas cerca de una casa con la puerta abierta y una luz encendida.
Entrar a la casa

Al entrar reconoces que es la casa del dios Thor y su esposa Sif, la cual parece estar en su habitación.
```

Obsérvese como al escribir una acción lógica correctamente hemos avanzado en la historia. Si por el contrario escribimos una acción sin sentido o no escrita correctamente (ya sea por palabras incorrectamente escritas o falta de las mismas), la aplicación puede que no reconozca lo que el usuario quiere decir y si esto pasa alertará de esto y volverá a imprimir el último texto impreso. A continuación mostramos un ejemplo:

```
Bienvenido a Loki Text Adventure

Capítulo 1
Eres Loki, Señor del Caos, hijo de los gigantes Farbauti y Laufey. Se te conoce hasta ahora como conflictivo, egocéntrico y megalómano. Tu actuación en esta historia llevará a que reine la paz en el mundo o este se sumerja en un caos total. Tienes tres hijos el enorme lobo Fenrir, la diosa de la muerte Hela y la serpiente marina Jormundgander. Tu historia comienza en Asgard, estas caminando por el medio de la plaza cuando pasas cerca de una casa con la puerta abierta y una luz encendida.
Entar a la casa

Acción inválida o irreconocible. Quizás faltan palabras o la acción no es esperada. Chequea también la ortografía de las palabras. No avance
Bienvenido a Loki Text Adventure

Capítulo 1
Eres Loki, Señor del Caos, hijo de los gigantes Farbauti y Laufey. Se te conoce hasta ahora como conflictivo, egocéntrico y megalómano. Tu actuación en esta historia llevará a que reine la paz en el mundo o este se sumerja en un caos total. Tienes tres hijos el enorme lobo Fenrir, la diosa de la muerte Hela y la serpiente marina Jormundgander. Tu historia comienza en Asgard, estas caminando por el medio de la plaza cuando pasas cerca de una casa con la puerta abierta y una luz encendida.
```

Fíjese que cuando salió el cartel por primera vez el usuario escribió mal la palabra `.Entrarz` por este motivo la acción no se pudo reconocer.

También durante el juego aparecerán situaciones en las que el usuario tendrá que escoger entre varias opciones. De esta forma la aplicación posibilita que el usuario pueda transitar por diferentes pasajes en base a sus decisiones. Cuan larga pueda ser la experiencia del jugador, también dependerá de las decisiones que tome. Una "mala" decisión, por decirlo de alguna forma, podría provocar que el usuario terminará el juego antes de terminar los 3 capítulos. Veamos un ejemplo donde el usuario se enfrenta a una decisión:

```
Thor se da cuenta que llevas la mayor parte de la cabellera de Sif escondida en una bolsa. Ahora tienes tres opciones : convencer a Thor de que puedes arreglar el daño que has hecho, huir de él o enfrentarlo.
```

En este caso el jugador podrá escoger entre convencer, hablar o enfrentar a Thor. En dependencia de lo que teclee el usuario se moverá a uno u otro pasaje.

4. Implementación

4.1. Estrategia de implementación

Como toda historia, esta está compuesta por diferentes pasajes. Por este motivo definimos un tipo "passage" para representar los diferentes pasajes. Más adelante veremos como está implementado este tipo. En todo momento el jugador se va encontrar transitando en un determinado pasaje. Para cambiar de pasaje, lo cual provoca un avance en la historia, el jugador teclea una sentencia, a la cual se realiza un análisis para determinar si se cambia a algún otro pasaje o si la historia se mantiene en el mismo. Cuando se alcanza algún pasaje "final" la ejecución de la aplicación se detiene.

4.2. Módulos implementados

La implementación de la aplicación, para mejor organización y comprensión, está dividida en los siguientes módulos: "Passages", "Synonyms", "Functions" y "Loki-Text-Adventure". En el módulo "Passages" está implementada la definición de el tipo "Passage", el módulo "Functions" se está la implementación de las funciones que permiten llevar el pasaje actual, el análisis de la entrada del usuario y el cambio o no de un pasaje a otro. Finalmente está el módulo principal "Loki-Text-Adventure" que apoyándose de los otros módulos permite el funcionamiento de la aplicación. A continuación veremos estos módulos por separado

4.2.1. Módulo Passages

Como se mencionó anteriormente, en este módulo está definido el tipo "Passage" del cual veremos ahora su definición.

```
data Passage = Passage { pid :: Int, text :: String, keywords :: [String], nextPossiblePassages :: [Passage]}
```

El entero "pid" lo utilizamos para comparar pasajes de una forma eficiente. El string "text" es el texto correspondiente a un pasaje. Esto es lo que se imprime en pantalla cuando se llega a dicho pasaje. La lista de Passage "nextPossiblePassages" son los pasajes a los cuales se puede avanzar a partir del pasaje actual. La lista de string "keywords" son las palabras que debe teclear el usuario (o algunas que sean sinónimos de estas o que transmitan una idea similar) que permiten a partir de un pasaje específico alcanzar el pasaje con esas keywords. O sea si yo el usuario se encuentra en el pasaje i y en la lista "nextPossiblePassages" del pasaje i se encuentra el pasaje j , par que el usuario pueda pasar del pasaje i al pasaje j deberá teclear las palabras que aparecen en la lista "keywords" del pasaje j (o algunas que sean sinónimos de estas o que transmitan una idea similar). De esta forma es que ocurre el cambio entre pasajes.

De la forma antes descrita, aparecen además definidos todos los pasajes de la historia. El pasaje inicial es el único que tiene su lista "keywords" vacía pues este no se alcanza desde ningún otro pasaje. Los pasajes finales, o sea los que no permitan avanzar al jugador más allá de ellos, se caracterizan porque su lista de pasajes "nextPossiblePassages" es una lista vacía.

4.2.2. Módulo Synonyms

4.2.3. Módulo Functions

En este módulo se encuentran definidas las funciones que permiten el correcto funcionamiento de la aplicación.

La función "actualPassage" permite la creación de una variable global en la que se guardará en todo momento el pasaje en el cual se encuentra la historia.

La función "changePassage" recibe como parámetro un "Passage" y lo que hace es definir al pasaje actual como el pasaje que se le pasa de entrada.

La función "obtainPassage" se utiliza para poder obtener el pasaje actual en el que se encuentra la historia.

4.2.4. Módulo Loki-Text-Adventure

Este es el módulo principal de la aplicación. En este módulo primero tenemos dos funciones "isWindows" y "setEncoding". La primera la utilizamos para saber si la aplicación se está ejecutando en Windows o no. La segunda es se utiliza para cambiar en el formato de codificación de caracteres a UTF-8 en caso de que no estemos en Windows. Esto lo hacemos porque nos paso que uno de nuestros programó en Windows 10 y el otro en Ubuntu 18.04 y si no cambiamos el formato de codificación de caracteres en Ubuntu la aplicación no reconocía los caracteres del Español como las tildes y la ñ. Sin embargo si cambiamos esta codificación en Windows ocurría entonces que dichos caracteres no se reconocían y al no cambiarlo si lo hacían. Por tanto se optó por comprobar si se estaba ejecutando la aplicación en Windows para cambiar o no la configuración de codificación de caracteres.