

Proyecto POUEx. Generador de juegos sobre mascotas virtuales

IMPORTANTE

- Se deben cumplir todos los requisitos que se indican en este documento.
- El proyecto será desarrollado utilizando las herramientas de desarrollo de software dirigido por modelos revisadas en esta asignatura.
- Es responsabilidad del alumno la custodia y protección de su proyecto.

Debido a la proliferación de los dispositivos móviles (teléfonos móviles o tabletas) entre los más pequeños de la casa, es habitual encontrar aplicaciones de entretenimiento basadas en el cuidado de una mascota virtual, siguiendo los pasos del popular tamagotchi de finales de los 90. Algunas de estas aplicaciones actuales podemos encontrarlas en el siguiente listado <http://www.android.es/los-mejores-juegos-de-tamagotchi-para-android.html> (que no puede considerarse completo). Un ejemplo muy interesante de este tipo de aplicaciones es el actual POU, disponible tanto para IOS como para Android.

El objetivo de este proyecto consiste en definir un lenguaje específico de dominio que permita, por una parte, el modelado de un juego de este tipo y, por otra parte, la generación del código necesario para producir dicho juego para una tecnología concreta. Esta tecnología concreta puede ser muy variada, desde una aplicación para dispositivos móviles, pasando por una aplicación de escritorio tanto textual como gráfica.

Para llevar a cabo este proyecto, será necesario la definición de un metamodelo que permita modelar el ciclo de vida de una mascota virtual, incluyendo los estados en los que la mascota puede encontrarse (sucio, feliz, triste), acciones (comer, limpiar, medicarse, etc.) y características (físicas, emocionales, etc.). Por una parte el metamodelo permitirá la definición de múltiples características físicas. Así, el modelador de la aplicación de mascotas puede definir conceptos como peso, altura, etc. Por otra parte se debe tener en cuenta la definición de características emocionales (tristeza, felicidad, sucio, etc.). Por último, será necesario definir acciones del ciclo de vida de la mascota, incluyendo la definición de las distintas situaciones (acciones) y cómo estas pueden afectar tanto a las características físicas como emocionales de la mascota virtual. Por ejemplo, si una mascota realiza la acción de comer estará saludable y, por ejemplo, puede aumentar de peso y altura.

En este proyecto se solicita el desarrollo de un lenguaje específico de dominio que facilite la definición y generación de juegos basados en mascotas virtuales. Para ello, el grupo de estudiantes deberá:

- Definir la sintaxis abstracta y restricciones OCL del lenguaje.
- Definir la sintaxis concreta del lenguaje, bien textual, bien gráfica.
- Construir los generadores de código necesarios para producir el juego propuesto a partir de los modelos definidos.
- Documentar adecuadamente cada una de las fases del desarrollo del proyecto.

El desarrollo de este proyecto incluye tres fases:

1. Definición de la sintaxis abstracta y semántica del dominio.

2. Definición de la sintaxis concreta del lenguaje específico de dominio. Esta sintaxis concreta puede ser textual o gráfica.
3. Definición de las transformaciones modelo a código que permitan la producción de los juegos modelados.

Sintaxis abstracta y semántica del lenguaje

La sintaxis abstracta de este lenguaje específico de dominio se definirá utilizando EMF (Eclipse Modeling Framework). Ésta definirá todos los conceptos y relaciones importantes para facilitar la posterior generación de código.

Queda a discreción del grupo de trabajo la identificación de los conceptos y relaciones necesarios para definir el lenguaje.

No obstante, deben tenerse en cuenta algunos aspectos:

- Deben existir tanto características físicas como características emocionales.
- Aunque será configurable, las características físicas básicas pueden ser: peso, altura, raza de la mascota, suciedad, etc. Del mismo modo, las características emocionales básicas son: felicidad, tristeza, salud, etc.
- Todas las características tendrán establecidos, al menos, los siguientes atributos: el valor inicial y valor máximo, si la característica es positiva o negativa. Por ejemplo, para la característica felicidad podríamos indicar: valor inicial = 5, valor máximo = 10, característica positiva (valores altos significan que la característica se acentúa).
- Las acciones en general y el tiempo, como una acción en particular, influyen positiva o negativamente en múltiples características, aumentando o disminuyendo el nivel o valor de las mismas. Por ejemplo, una acción denominada jugar podría influir positivamente en la felicidad y negativamente en la tristeza. No obstante, la acción jugar podría estar condicionada por el tipo de juego. Por ejemplo, jugar a la pelota en compañía puede ser más divertido que jugar solo/a y, por tanto, jugar a la pelota puede incrementar en mayor medida el nivel de felicidad que la acción jugar solo/a. Entre las acciones que podemos definir se incluyen: comer, limpiar, jugar, dormir, medicarse, etc.
- El transcurso del tiempo puede influir positiva o negativamente sobre alguna o todas las características.
- La mascota puede estar en distintos estados, los cuales dependen de valores concretos o combinaciones de valores concretos de ciertas características. Por ejemplo, la mascota estará sucia si el valor de la característica suciedad es mayor que 7. Así, será necesario modelar los estados y las condiciones de activación de esos estados. Nótese que una mascota puede estar en múltiples estados, por ejemplo: sucio y enfermo.
- El lenguaje puede definir valores críticos de alguna característica tanto física como emocional que permita identificar la finalización del juego. Por ejemplo, tristeza = 1, esto podría significar que la mascota termina muriendo de tristeza.

Restricciones asociadas a la semántica del lenguaje

Las restricciones que se indican a continuación deben codificarse en OCL para restringir la sintaxis abstracta definida. Queda a elección de grupo de trabajo su ampliación.

- Todas las características tendrán establecidos el valor inicial y valor máximo, si la característica es positiva o negativa.
- El conjunto de acciones deben modificar el conjunto de características definidas, es decir, todas las características se verán afectadas por alguna acción
- Una acción no puede modificar dos veces el nivel de una determinada característica.

Sintaxis concreta.

El lenguaje específico que se diseñe debe disponer de una sintaxis textual o gráfica. En ambos casos, será necesario desarrollar los plugins Eclipse correspondientes: bien utilizando Xtext, en el caso de sintaxis textual o GMF/Eugenia en el caso de sintaxis gráfica.

Independientemente de la sintaxis concreta definida para el lenguaje, en la documentación del DSL se incluirá la información necesaria para facilitar el desarrollo de modelos basados en esta sintaxis concreta, incluyéndose los ejemplos utilizados durante el desarrollo.

Generación de código

El código a generar debe incluir todos los aspectos necesarios e imprescindibles para producir el ejecutable del juego definido. En este documento no se define una tecnología concreta como objetivo de la generación de código. Por tanto, queda a elección del grupo de trabajo la selección de la tecnologías o tecnologías objetivo del mismo. Por ejemplo, podría generarse código para dispositivos móviles, para páginas Web, para una aplicación de escritorio utilizando alguna librería gráfica o generar una aplicación de consola basada en texto.

No obstante, independientemente de la tecnología utiliza para representar la información se deben tener en cuenta las siguientes restricciones:

- Para generar el código se utilizará **Acceleo** como lenguaje de transformación de modelos a texto.
- Se mostrarán los valores concretos de cada una de las características de la mascota.
- Se mostrará el estado actual de la mascota. La forma de mostrar el estado actual variará en función de la tecnología concreta elegida: textual, gráfica, etc.
- El usuario podrá seleccionar las acciones disponibles para interactuar con la mascota.

Extensiones de generación de código adicionales

A continuación se definen algunas características opcionales del proyecto, que se calificará aparte de generación de código presentada previamente, suponiendo **hasta 3 puntos adicionales** a la calificación del proyecto.

- **Generación de información persistente de jugadores y estado del juego.** El código generado mantendrá información persistente del juego. Es decir, será posible acceder/recuperar el estado de la mascota de un usuario concreto.
- **Generación de un entorno gráfico para el juego.** La extensión concreta incluye la generación de un entorno de juego gráfico. Queda a elección del grupo de estudiantes la selección de la tecnologías y librerías a utilizar.
- **Generación de la aplicación para un dispositivo móvil.** Esta extensión incluye la generación de código para una plataforma móvil, por ejemplo, Android, IOS, etc.

Documentación del proyecto

(ver plantilla de documentación)

Entrega

Este proyecto se realizará en grupos de como **máximo tres personas**. Los grupos de trabajo deben documentar las responsabilidades de cada uno de los miembros y la evolución del desarrollo del proyecto en un documento que, posteriormente, anexarán a la documentación del proyecto.

Fecha de entrega: *Martes 6 de mayo antes de las 23:00* a través del campus virtual.

Se entregará un archivo comprimido **dmss_pouex_proyecto_grupoXX.zip** que incluirá:

- Todos los proyectos eclipse implementados y generados (syntaxis abstracta, syntaxis concreta, generador de código).
 - dmss.pouex
 - dmss.pouex.edit
 - dmss.pouex.editor
 - dmss.pouex.test
 - dmss.pouex.diagram ó dmss.pouex.xtext
- Proyectos de ejemplo generados con el DSL desarrollado.
 - dmss.pouex.webapplicationgen
- Documentación del proyecto
 - dmss_pouex_documentación_grupoXX.pdf