

Medieval Game. Diagrama UML de clases de un juego interactivo.

30 de enero de 2020

1. Tarea

Tu tarea consiste en elaborar el **diagrama de clases** y la **documentación** para el análisis de una aplicación que implemente el juego de rol descrito más abajo con la aplicación Visual Paradigm. Hay que **generar el código** de la aplicación. Si se os acaba el tiempo de evaluación tenéis otras alternativas, como dia2code o BOUML. Al final debe haber un proyecto NetBeans que integre ese código. Para documentar el proceso deberás entregar un documento de texto con los siguientes puntos:

- **Extracción de los sustantivos** en la descripción del problema.
- **Selección** de *sustantivos* como **objetos/clases** del sistema.
- Obtención de los **atributos** de los objetos.
- Obtención de los **métodos**.
- Obtención de las **relaciones**.
- Añadir **getters, setters y constructores**.
- Primer **refinamiento**.
- **Documentación**.

1.1. Criterios de puntuación. Total 10 puntos.

- Crear la *tabla de sustantivos y categorizarlos, seleccionar las clases*. **2 puntos**.
- Obtención de las *relaciones*. **1 punto**.
- Obtener los *atributos*. **1 punto**.
- Obtener los *métodos*. **1 punto**.
- Crear el *diagrama*. **1 punto**.
- Añadir *getters, setters y constructores*. **1 punto**.
- *Refinar* el diagrama. **1 punto**.
- Añadir *documentación*. **1 punto**.
- Hacer el *proyecto en NetBeans | VSCode | PyCharm... , StarUML | Visual Paradigm* y *generar el código*. **1 punto**.

2. Descripción

2.1. Medieval Game

Se propone la realización de un juego de rol. El juego estará basado en la ficción y se sitúa la historia en la edad media. El usuario tiene que crear y mantener una ciudad tomando una serie de decisiones. El objetivo es hacer prosperar la ciudad lo más posible. El juego tendrá un motor de ejecución que irá modificando las condiciones del juego y hará que este sea factible de jugar.

La ciudad que el usuario quiere construir tiene una serie de ciudadanos:

2.1.1. Ciudadanos:

- Granjeros. Producen comida para la población. Los granjeros suelen tener descendencia.
- Artesanos. Producen armaduras, espadas y otros elementos para el ejército. Los artesanos suelen tener descendencia.
- Militares. Luchan pero tienen que estar debidamente armados. Si no tienen armas son soldados indefensos que mueren cuando viene el ataque bárbaro. Si tienes muchos militares y pocos granjeros no hay comida para la población. Los militares no tienen descendencia dado que siempre están en escarceos y patrullas militares. Puede haber militares de diverso tipo, dependiendo del tipo de arma que utilizan. Se organizan en compañías, compuestas por diversos tipos de militares. Según la organización de cada compañía se enfrentarán mejor o peor a los ataques.
- Taberneros. Dirigen una taberna. Necesitan X artesanos para que hagan el hidromiel. No tienen descendencia puesto que siempre están trabajando en la taberna. Una taberna puede dar a basto a X ciudadanos (excluyendo a los curas que no beben).
- Curas. Hacen gasto básicamente y defienden a la población de la aparición de falsos profetas.

La aplicación debería mostrar al usuario el número de cada uno de los ciudadanos.

2.1.2. Eventos:

- Sequía. La producción de los granjeros se reducirá un x %
- Ataque Bárbaro. Si no tienes bastantes militares te van a matar a más ciudadanos. Dependiendo del % de militares el número de víctimas puede ser mayor o menor. El % de la población que pierdes varía dependiendo del % de militares que tengas. Pueden ser de distinto tipo.
- Falsos profetas. Si no tienes suficientes curas los falsos profetas embaucan a la población y hacen que los granjeros y artesanos peregrinen y pierdes población.
- Revuelta callejera. Si no tienes suficientes tabernas los artesanos y los granjeros se amotinan. Tienes que llamar a los militares y pierdes un x % de la población.
- Peste. Perderás un % de la población dividida en x jornadas.
- Nacimiento. En cada jornada de cada X habitantes nacerá un nuevo ciudadano.
- Recaudar. Según quiera el jugador se puede pedir impuestos a la población. Con los impuestos se puede hacer lo siguiente:
 - Montar una taberna. Con lo cual la felicidad de la población aumenta.
 - Montar un espectáculo. Con lo cual la felicidad de la población aumenta.

Con los impuestos se enfadan los granjeros y artesanos y aumenta la probabilidad de una revuelta callejera. La aplicación tiene que visualizar la alegría de la población.

2.1.3. Sistema interactivo.

El sistema te va a ir haciendo preguntas y dependiendo de las respuestas dadas hará algo. Es la base del juego. Sistema multiusuario y Login

El sistema es multiusuario y tiene que tener un sistema de logueo a la aplicación.

Se considerará un plus que se recuerde el usuario durante un tiempo determinado en el mismo equipo y que no haga falta volver a introducir la clave. No se almacenarán claves en claro solamente en formato MD5.

2.1.4. Motor.

Va por jornadas. El sistema comienza en el año 476 D.C. En cada jornada ocurre uno o varios eventos de forma aleatoria.