

Programación Orientada a Objetos
Proyecto Unidad 1
15 de septiembre de 2016

Fecha primera parte: A más tardar el jueves 29 de septiembre de 2016, a las 14:30 horas

Fecha segunda parte: A más tardar el lunes 03 de octubre de 2016, a las 12:00 horas

Descripción

Como jugadores o desarrolladores de videojuegos nosotros sabemos que la inmersión lo es casi todo. Cuando estás inmerso tu pierdes la noción del tiempo y te involucras en lo que el videojuego te está presentando. Factores muy importantes a considerar cuando se diseña la experiencia inmersiva son, qué tan fluida es la experiencia del usuario o qué tan bien diseñada es la interfaz de usuario. Un mal diseño de interfaz de usuario puede hacer que la experiencia de uso de videojuego sea muy mala.

Es por lo mencionado anteriormente, que lo han contactado a Ud. y a su grupo para el desarrollo de un sistema computacional que ayude a la construcción de una interfaz de usuario. Este sistema debe proporcionar todas las herramientas necesarias para incorporar elementos gráficos a una interfaz de usuario y así posteriormente generar un archivo de texto el cual contenga la información de todos los elementos gráficos.

Todos estos elementos que se incorporarán al sistema serán recursos gráficos (*Sprites*). Para cada recurso gráfico, el sistema deberá almacenar la siguiente información:

- Identificador del recurso, este valor es proporcionado por el usuario y deberá ser único.
- Ruta del archivo que contiene el recurso gráfico en el computador.
- Tamaño del recurso, en específico el alto y ancho en píxeles.
- Tipo de archivo, el cual puede ser raster o vectorial.

Con los recursos gráficos cargados en el sistema, una escena (*Stage*) será el agrupamiento de varios recursos gráficos dispuestos en algún lugar de la escena. La información básica de una escena será la siguiente:

- Nombre de la escena. Este nombre no podrá repetirse en las escenas registradas en el sistema.
- Tamaño de la escena, en específico el alto y ancho de la escena en píxeles.
- Lista de los recursos gráficos utilizados en esta escena.

Cada escena tendrá un sistema coordenado, el cual comenzará en la esquina inferior izquierda y aumentará hacia la derecha en el eje X y hacia arriba en el eje Y. Cada recurso perteneciente a una escena deberá estar contenido en el área de la escena, es decir, su origen deberá estar

dentro del área comprometida por la escena. Cada recurso dentro de una escena deberá contener la siguiente información:

- Identificador del recurso en la escena. Este valor será definido por el sistema, debe ser único y puede ser distinto del identificador general del recurso.
- Ubicación del recurso dentro de la escena, en específico las coordenadas (x, y) dentro de la escena.
- Tamaño del recurso en la escena, cabe señalar que este tamaño puede ser distinto del tamaño original del recurso.
- Ángulo de rotación del recurso en la escena, valor mayor o igual que 0° y menor que 360°.
- Etiqueta del recurso en la escena, cabe señalar que varios recursos en una misma escena pueden tener asociada una misma etiqueta.

Adicionalmente, a un recurso de una escena se le podrá asociar un polígono para la detección de colisiones. Este polígono consiste en una lista de puntos en el plano 2D, donde cada punto está conectado con el siguiente punto en la lista y, además, el último punto está conectado con el primero.

Entonces, una vez que los recursos gráficos están cargados al sistema y ya se han agregado las distintas escenas que contendrá el videojuego, estas se podrán exportar a un único archivo XML, el cual deberá tener un formato preestablecido. A continuación, se presenta un ejemplo:

```
<?xml version="1.0" encoding="UTF-8" ?>
<World>
  <SpriteList>
    <sprite>
      <Id>R001</Id>
      <Path>c:\UI\resources\images\img01.png</Path>
      <Width>30</Width>
      <Height>30</Height>
      <Type>RASTER</Type>
    </sprite>
    <sprite>
      <Id>R002</Id>
      <Path>c:\UI\resources\images\img02.png</Path>
      <Width>1366</Width>
      <Height>768</Height>
      <Type>RASTER</Type>
    </sprite>
  </SpriteList>
  <StageList>
    <Stage>
      <Name>Main Stage</Name>
      <Width>1366</Width>
      <Height>768</Height>
      <Resources>
        <Resource>
          <Id>MS001</Id>
          <SpriteId>R001</SpriteId>
          <X>0</X>
          <Y>0</Y>
          <Width>1366</Width>
          <Height>768</Height>
          <RotationAngle>0</RotationAngle>
          <Tag>Background</Tag>
        </Resource>
        <Resource>
          <Id>MS002</Id>
          <SpriteId>R002</SpriteId>
```

```

        <X>50</X>
        <Y>200</Y>
        <Width>30</Width>
        <Height>30</Height>
        <RotationAngle>0</RotationAngle>
        <Tag>Player<Tag>
        <Polygon>
            <Point>
                <X>0</X>
                <Y>0</Y>
            </Point>
            <Point>
                <X>0</X>
                <Y>30</Y>
            </Point>
            <Point>
                <X>30</X>
                <Y>30</Y>
            </Point>
            <Point>
                <X>30</X>
                <Y>0</Y>
            </Point>
        </Polygon>
    </Resource>
</Resources>
</Stage>
</StageList>
</World>

```

Requisitos

Su sistema deberá persistir la información al cierre de la aplicación. Es decir, el sistema al volver a ejecutarse, posterior a un cierre normal, deberá contener toda la información previamente cargada. Cabe señalar que no es necesario que utilice el mismo formato descrito anteriormente.

Su sistema deberá contar con un menú de usuario, el cual permita realizar las siguientes tareas:

- Comenzar nuevo proyecto. Con esta opción el sistema deberá limpiar todos los datos previamente almacenados.
- Agregar, editar y eliminar:
 - Recursos gráficos al proyecto
 - Escenas
 - Recursos a una escena
 - Polígonos para la detección de colisiones a un recurso de una escena
- Listar los recursos gráficos del proyecto
- Listar las escenas
- Listar los recursos gráficos de una escena
- Mover recursos gráficos dentro de una escena, especificando su Id o su Etiqueta. Esta acción funciona como una traslación, preguntándole al usuario cuanto quiere mover el elemento en cada eje.

Adicionalmente, al momento de generar los listados, el usuario deberá seleccionar si quiere que este listado aparezca en el orden en que fueron agregados los elementos, en orden ascendente o descendente. Para los listados en orden ascendente o descendente deberá consultarle al usuario por qué campo los quiere ordenar, por ejemplo, para generar el listado de escenas podría elegir ordenarla ascendentemente usando como criterio el nombre.

Su sistema deberá manejar todas las excepciones que puedan ocurrir de manera que su sistema no tenga cierres abruptos. Por último, no podrá usar librerías externas para la lectura y escritura de archivos, menos para la creación del archivo XML, ni tampoco podrá usar las librerías XML de java para este propósito.

Forma de trabajo y entregas

Este proyecto debe ser desarrollado en grupos de 3 estudiantes. Existen dos entregas. El estudiante recibirá retroalimentación después de la primera de ellas. La primera entrega consta de un informe y de un prototipo de las clases. En el informe se describen las clases de objetos definidas para resolver el problema. Cada clase debe especificarse formalmente y acompañarse de una descripción de los métodos y atributos de clase (variables de instancia) definidos. Asimismo, debe explicarse la forma en que, utilizando las clases definidas, se van a implementar las distintas funcionalidades solicitadas en los requerimientos. Esto debe hacerse en forma exhaustiva, es decir, abarcando todas las funcionalidades. El prototipo debe contar con los principales métodos, ya sean constructores, accesores y mutadores, que permitirán implementar las funcionalidades solicitadas. La implementación de algunos métodos que revistan especial complejidad podrá postergarse para la segunda entrega. El prototipo no debe contener programa principal.

La segunda entrega incluye el programa completo y un informe que considera la etapa anterior, con las modificaciones que fue necesario hacer para cumplir adecuadamente con los requerimientos. Debe incluir un diagrama de clases. Además, debe contener los resultados de las pruebas llevadas a cabo y una descripción de las limitaciones o restricciones que tiene el programa, incluyendo comentarios sobre la forma de solucionarlas.

Todas las entregas, tanto los informes como las que involucran código, deben hacerse a través de Educandus. Los informes deben entregarse en formato PDF. El código fuente debe entregarse en un único archivo zip.

Evaluación

Los porcentajes de evaluación del proyecto descrito en el Syllabus son los siguientes:

- Documentación inicial de clases (20%)
Corresponde a la primera entrega – informe
- Prototipo del proyecto (20%)
Corresponde a la primera entrega – prototipo
- Entrega final proyecto (30%)
Corresponde a la segunda entrega – código fuente
- Documentación final de clases (5%)
Corresponde a la segunda entrega – informe
- Documentación de código (5%)

Corresponde a la segunda entrega – JavaDoc