

Ecosistemas de Aplicaciones.

2019-2

Taller Individual 1

Juego con control remoto

Introducción.

Deberá crear una aplicación que sea un juego y permita a los usuarios controlarlo desde dos dispositivos (1 móvil y 1 fijo o móvil), permitiendo que cada usuario interactúe con un equipo separado. El juego puede ser del tema que se elija, debe tener una pantalla de inicio, instrucciones y debe permitir que lo que realice el usuario sea controlado desde un dispositivo móvil. Este desarrollo deberá contar con tres pantallas: Interfaz de juego (Escritorio), Control jugador 1 (móvil o escritorio) y jugador 2 (móvil), siendo la pantalla de escritorio la que tiene el diseño visual de la aplicación y el dispositivo móvil funcionará como un control que permitirá generar los movimientos y acceder a otras funcionalidades como botón de ayuda o de reinicio o selección y navegación. **Sólo se puede usar el mouse para dar inicio a la aplicación de escritorio.**

Debe manejar límite de tiempo y puntaje.

Propuesta de Diseño:

- Título del proyecto.
- Conceptualización
- Referencias teórica y gráficas.
- Propuesta gráfica de la interfaz y elementos de interacción.
- Explicación detallada de la interacción.

Condiciones de interacción y funcionamiento:

Programación.

- El taller deberá desarrollarse en Eclipse y Android Studio.
- El código entregado debe ser funcional: debe compilar sin errores, no debe requerir cambios en el mismo con excepción de la dirección IP (en el caso de la aplicación de Android debe permitir seleccionar dentro de la interfaz la ip a la cual se conectará) y debe tener los archivos requeridos para su funcionamiento en Eclipse (Esto incluye imports y demás elementos).
- El taller será individual.
- **Es requisito que se presente el taller de manera presencial durante la práctica para poder ser evaluado.**

Diseño.

- Debe ser claro lo que debe hacer el usuario en la interacción.
- Deben estar conectados los dos equipos para que funcione el juego (escritorio y Android)
- Debe tener control de tiempo.
- Debe tener control de puntaje.
- El tamaño del lienzo no puede ser mayor a 1200*700 px.

- Es requisito que se presente el taller de manera presencial durante la práctica para poder ser evaluado.

Cronograma fechas de entrega esperada:

Recordar que la entrega máxima es hasta 4 días después de la entrega esperada hasta las 11:55 pm (2 días sin penalidad)

Propuesta de Diseño en behance: **Clase 1 de la semana 8.**

Descripción de los componentes y diagrama de clases UML en github Readme: **Clase 2 de la semana 9.**

Código final: **10 octubre (23:55) (Semana 11) - Moodle**

Revisión del código final: **15 de octubre (durante la clase) (semana 12)**

Criterios de evaluación

Análisis 15%: Diagrama de clases y documentación en sketch.	
Diagrama de clases: Existe la clase Ejecutable con atributos y métodos estáticos, y sin relaciones con otras clases.	4%
Diagrama de clases: Las clases que solucionan el problema están definidas y diagramadas correctamente (incluyendo nombres en singular, estándar de nombre para Lógica y Ejecutable). Requisito: Están definidas todas las clases acorde a lo requerido por el enunciado y la propuesta.	4%
Diagrama de clases: Las clases que solucionan el problema, tienen los atributos y métodos adecuados y correctamente diagramados (Visibilidad, retorno, parámetros, tipos y nombres), sin atributos que representan relaciones. Requisito: Están definidas todas las clases acorde a lo requerido por el enunciado y la propuesta.	4%
Diagrama de clases: Están correctas todas las relaciones necesarias entre las clases (direccionalidad, cardinalidad, tipo de relación y diagramación). Requisito: Están definidas todas las clases y relaciones acorde a lo requerido por el enunciado y la propuesta.	4%
Documentación en sketch: Está la descripción de funcionalidad acorde al esquema de los atributos, variables globales y métodos del diagrama de clases (excepto get y sets). Requisito: Están definidas todas las clases y relaciones acorde a lo requerido por el enunciado y la propuesta.	4%

Código 60%.	
Correspondencia: Existe adecuada correspondencia entre el enunciado del taller, la propuesta y la solución. Requisito: Están definidas todas las clases acorde a lo requerido por el enunciado y la propuesta. Requisito: Están definidas todas las relaciones acorde a lo requerido por el enunciado y la propuesta.	5%
Utilización correcta de las clases y los objetos, estos se referencian desde las clases adecuadas, no se accede directamente a los atributos desde otras clases y no se traen los objetos a la ejecutable. Existe una separación adecuada de los métodos asociados a crear, dibujar, enviar, etc. Se utiliza la clase Lógica o equivalente en la solución del problema. Requisito: Están implementadas todas las clases acorde a lo requerido por el enunciado y la propuesta.	10%
Código Funcionamiento de los hilos para la búsqueda e identificación de nodos. Requisito: Está implementada correctamente la lógica y la clase encargada de buscar los nodos.	5%
Código Funcionamiento de la aplicación para reiniciar y observar las propiedades. Requisito: Está implementada correctamente la lógica y la clase encargada de buscar los nodos.	10%
Creación efectiva de los elementos producto de la búsqueda de nodos, desaparición al perder una dirección en la red, se piden los datos secuencialmente y usando un hilo. Requisito: Está implementada correctamente todas las clases.	10%
Funciona correctamente la aplicación en su conjunto, existe interacción efectiva y corre fluidamente mediante el uso adecuado de hilos. Requisito: La aplicación corresponde completamente al enunciado y la propuesta.	10%

Es creativa la propuesta gráfica y visual de la aplicación, muestra investigación y profundización en su propuesta. Requisito: Entrega de la ficha con referencias teóricas y gráficas con bibliografía completa. Requisito: La ficha técnica no presenta errores de ortografía.	5%
Creatividad en la interacción del usuario, es clara la interacción y es fácil de usar. Requisito: Todas las funciones del juego están en el dispositivo móvil.	5%
Coherencia en el desarrollo de la interfaz Requisito: Muestra controles de tiempo y puntaje.	5%
Proyecto documentado y disponible en Behance	10%

Esquema de documentación de un método

```
/**
 * Descripción corta, para todos los métodos (excepto los get/set)
 *
 * Descripción completa, para los métodos no triviales (que tienen muchas
 instrucciones)
 *
 *
 * @parametro      tipo      nombre_parametro      Descripción
 * @retorno        tipo      Descripción
 */
```

Esquema de documentación de un atributo/variable global

```
tipo      nombre_atributo      // Descripción de la utilidad del atributo/variable global
```