

**Asignatura:**

**ST-ISC210-T-001 Programación Aplicada**

**Trabajo de Curso:**

**2012-0467 Manuel Tolentino**

**2013-1459 Jean Louis Tejeda**

**Autores:**

**2012-0467 Manuel Tolentino**

**2013-1459 Jean Louis Tejeda**

**Tutor:**

**Ing. Miguel Moronta**

**Entregado en Fecha:**

Miércoles, 18 de noviembre de 2015

**PONTIFICIA UNIVERSIDAD CATÓLICA MADRE Y MAESTRA – PUCMM  
FACULTAD DE CIENCIAS DE LAS INGENIERÍAS  
Departamento de Ingeniería en Sistemas y Computación – ISC**

# Resumen / Abstracto del Problema

Charlie Circus fue lanzado por la compañía Konami en 1984 para la NES; en este videojuego se interpreta a una estrella del circo que cumple con varias actuaciones para entretener al público.

Las actuaciones principales para esta entrega son:

* Atravesar aros de fuego a los lomos de un león.
* El salto entre ruedas en movimiento.
* Salto entre cuerdas.

En todas las actuaciones hay desplazamiento y cada una tiene más de lo que se ve en un primer plano, en cada una de ellas hay que recorrer varios metros para llegar al final de la misma, por lo que la cámara debe moverse para ir viendo cada retazo del escenario según el actor vaya avanzando.

También dado que muchas cosas se repiten en el escenario se implementara el diseño basado en celdas para construir los escenarios.

Aparte de lo que originalmente tenia Circus Charlie añadiremos distintos modos de dificultad (Fácil, Media, Difícil). Y permitiremos la selección del tipo de actuación que se quiere realizar.

Se desea implementar también aparte del manejo por teclado, el manejo con mandos/controles de consolas tales como el de Xbox 360 para dar una experiencia más cómoda.

Tabla de contenido

[Resumen / Abstracto del Problema 1](#_Toc409432871)

[Análisis 1](#_Toc409432872)

[Cuerpo del Informe 1](#_Toc409432873)

[Código Fuente 1](#_Toc409432874)

[Condiciones de Error 1](#_Toc409432875)

[Notas al Profesor 1](#_Toc409432876)

[Conclusiones 1](#_Toc409432877)

[Recomendaciones 1](#_Toc409432878)

[Referencias Bibliográficas 1](#_Toc409432879)

# Análisis

Para TCircus 2015 nuestro propósito es crear un juego con un personaje que se desplaza por un escenario en distintas condiciones, superando los obstáculos que se le presenten en su camino.

Para este proyecto se hará uso de cierto tipo de fórmulas para generar movimientos deseamos. Donde se pueden mencionar el movimiento rectilínea uniforme, el cual se implementó en el movimiento constante del personaje; también el movimiento rectilíneo uniformemente variado, el cual se implementa presionando una tecla que cambia la velocidad del personaje; otro que se puede mencionar es el movimiento parabólico, el cual se implemente en los saltos del personaje.

En esta primera entrega existen varios niveles como son fácil, medio y difícil. Una breve explicación de cada uno es:

1. Fácil: Los obstáculos se encuentran con una distancia manejable entre sí.
2. Medio: En este nivel todo sucede más rápido, ya que cada cierto tiempo la velocidad aumenta.
3. Difícil: Los obstáculos esta con menos distancia de separación y la velocidad es bastante alta, prácticamente un modo que muy pocas personas pueden lograr ganar.

Para esta primera parte también existen varios niveles, la cual nosotros llamamos distintos escenarios, que de igual manera cada uno tiene su relativa dificultad. Por ultimo este juego poseerá sus funcionalidades en una pantalla general el cual sea amigable con el controlador del mismo software y este pueda acceder a las distintas funcionalidades de dicho juego.

# Cuerpo del Informe

El desarrollo de un juego TCircus es una herramienta que creara una nueva forma de entretenimiento para sus usuarios.

La creación de dicho juego se basa en la clásica forma de divide y vencerás, separando cada uno de los elementos en estratos y dándoles sus respectivos valores de prioridad e importancia para así trabajar por separado en cada uno de los estratos, y al final crear una nuevo juego que le guste a los usuarios.

Las relaciones de las clases creadas para el proyecto han sido plasmadas en unos diagramas de clases UML, en los cuales se puede ver cuáles son las dependencias de cada uno de los objetos y clases desarrolladas para poder enfrentar a nuestro problema y llegar a una solución eficaz.

# Código Fuente

No aplica en esta ocasión.

# Condiciones de Error

No existen condiciones de error.

# Notas al Profesor

Juego sujeto a modificaciones en segunda entrega.

# Conclusiones

Dentro de los objetivos planteados para esta entrega se puede apreciar que cada uno fue analizado profundamente, pero por términos de tiempo no pudieron ser llevados a la implementación que se debía.

Se utilizaron varias técnicas para la eficiencia y agilización del desarrollo de la aplicación. Una de ellas fue dividir los problemas en varios más pequeños e irlos asociando uno con el otro, donde vivimos ciertas incoherencias ya que cada miembro pensaba una manera distinta en cómo solucionar el problema, la cual nos permitió no solo tener una opción sino que varias opciones e incluso implementar más de una opción a la vez. Esta parte nos ayudó a aprender como programadores e incluso a vivir experiencias que en un futuro tendremos.

Mientas que otra fue la implementación de programación por pares (pair programing), con un “piloto” programando y un “copiloto” dando indicaciones y verificando que siempre siguiéramos el diagrama de clases UML que nos planteamos para resolver el problema, cada ciertos lapsos de tiempo fuimos intercambiando nuestros puestos, para así cada quien hacer de su parte en cada ámbito. También por momentos dialogamos sobre problemas particulares en los que trabajáramos en ese instante, para ver cuál era la mejor y más eficiente forma de implementar en el sistema la solución de dicho problema.

Por último se implementó la herramienta llamada GitHub como una de las técnicas más importantes utilizadas, ya que en sesiones en las que el equipo no estaba reunido cada uno trabajo en partes asignadas y el otro podía ver estos cambios casi de inmediato e incluso hacer modificaciones a esta misma parte para perfeccionar la misma.

# Recomendaciones

No hay ninguna recomendación que apuntar.

# Referencias Bibliográficas

<http://docs.unity3d.com/es/current/Manual/UnityManualRestructured.html> - Documentación de Unity.