

**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, 30 de septiembre 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

Snake es un videojuego en el que el usuario controla la cabeza de una serpiente que se desplaza por un escenario con el objetivo de comer algunas frutas que le irán aumentando su longitud. La idea es que el usuario consiga la mayor longitud posible de la serpiente sin tropezarse con su propio cuerpo.

Ahora bien, Snake RoboVacumm 2015 se basa en unos robots de limpieza, llamados "La Serpiente 2015", que hacen su trabajo diario. En vez de comida se recoge basura, y en vez de partes-del-cuerpo/cola-creciente se tienen mini-bots. Existen varios modos de juegos: clásico, power-up, contra-reloj y contra-reloj extremo. Cada modo tiene múltiples niveles de dificultad: fácil, medio, difícil; a excepción del modo contra-reloj extremo que solo tiene dificultad masoquista.

En la partida se tienen opciones de pausa, repetir y salir. También se mostrara el puntaje actual y máximo. Desde el menú inicial se podrá acceder a una tabla con el top de puntuaciones.

Hay dos robo-serpientes, una verde y otra naranja, estas pueden competir en un modo multijugador, y en el modo power-up cada una tendrá una posibilidad distinta de obtener algún power-up específico.

En el modo clásico, aparte de lo dicho sobre Snake se tiene:

Bola de beisbol: Una bola se desplaza por el escenario [MRU], a veces rebotando con las paredes [Pasará a MRUV], si choca con el robo-serpiente la dejara inmóvil por un tiempo, mientras que si golpea a un mini-bot esté y todos los que estén detrás de él morirán.

Power-up es un modo en el que los robo-serpientes por medio de conexión inalámbrica pueden usar objetos del hogar para hacer frente a ciertos inconvenientes que aparecerán, los poderes serán principalmente:

Guante de beisbol: Captura la bola para que no haga daño al robo-serpiente.

Encender aire acondicionado: Congela la basura en su posición (en este modo la basura se pasea por el escenario [MRU]).

Batería sobrecargada: La robo-serpiente toma una batería sobrecargada que acelera su velocidad. [Pasa de MRU a MRUV]

En este modo también hay “Power-down”:

Fuego: La robo-serpiente se quema y pierde el control de sí misma por unos segundos. [MRU]

Descarga de energía: Hace a la robo-serpiente mucho más lenta. [Pasa de MRU a MRUV con desaceleración]

En el modo contra-reloj se desafiara al jugador con el estilo del modo clásico. El objetivo será ir limpiando para que el tiempo no caiga a 0 (Ósea, limpiar aumentará el tiempo que le sobra al jugador para que termine la partida).

Finalmente el modo contra-reloj extremo sumará al anterior los power-down (no power-up), y usará una dificultad muy superior a la de los otros modos.

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 Snake RoboVacumm 2015 nuestro propósito es crear un juego con una serpiente que se desplaza por un escenario encerrado comiendo lo que nosotros le llamamos basura, y según come aumenta su tamaño o longitud.

Dentro de este proyecto se utilizara el movimiento rectilíneo uniforme para que la serpiente vaya a una misma velocidad, así como el movimiento rectilíneo uniformemente variado, el cual se aplica cuando la serpiente come algún tipo de poder que le aumente su aceleración.

Ya 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 Snake RoboVacumm 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 y llevado a la implementación que se debía. Snake RoboVacumm cumple con cada uno de los requisitos previamente indicados.

Se utilizaron dos técnicas para la eficiencia y agilización del desarrollo de la aplicación. La primera fue dividir los problemas en varios más pequeños e irlos asociando uno con el otro. Esta parte facilito la parte del diseño y estructura del juego para llegar a los resultados esperados.

Mientas que la segunda 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.

# Recomendaciones

No hay ninguna recomendación que apuntar.

# Referencias Bibliográficas

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