

Instituto Tecnológico de Costa Rica	Primer Proyecto Programado
Departamento de Computación.	Robots – Linux
CE-1102 Taller de Programación	Prof: Ing. Jeff Schmidt Peralta
I Semestre 2014 3 de octubre de 2014	Consultas al correo: jschmidtcr@gmail.com

1. Introducción.

Algunos de los principales esfuerzos de programación se dedican a la elaboración de juegos de computadora o videojuegos. En la presente tarea se va a realizar la lógica básica de un juego de computadora implementando los conceptos de objetos, matrices y manipulación de imágenes.

En la presente tarea se va a realizar un juego MUY básico llamado ROBOTS, inspirado en el juego del mismo nombre desarrollado en la Universidad de Berkeley para los sistemas operativos UNIX y LINUX.

Existen diferentes versiones de este juego llamadas Zombies, Daleks, Robot MineField y muchas otras.

2. Descripción Funcional del Programa.

Todos los videojuegos comparten entre sí algunos aspectos básicos. Tiene un ciclo de juego (el proceso de repetir y repetir hasta que el juego acabe), al juego se le da una entrada, procesa la entrada, actualiza los elementos del juego, y verifica condiciones de victoria/fracaso.

Los principales elementos involucrados en la elaboración de video juegos son:

- Matemática básica
- Física básica
- Manipulación de objetos
- Manipulación de gráficos
- Sincronización de procesos
- Mapeo
- Detección de colisiones
- Inteligencia Artificial.

La presentación de la interfase con el usuario es **totalmente libre**, puede contener menús, botones, imágenes o cualquier otro componente que se desee utilizar y será un elemento importante dentro de la calificación del proyecto.

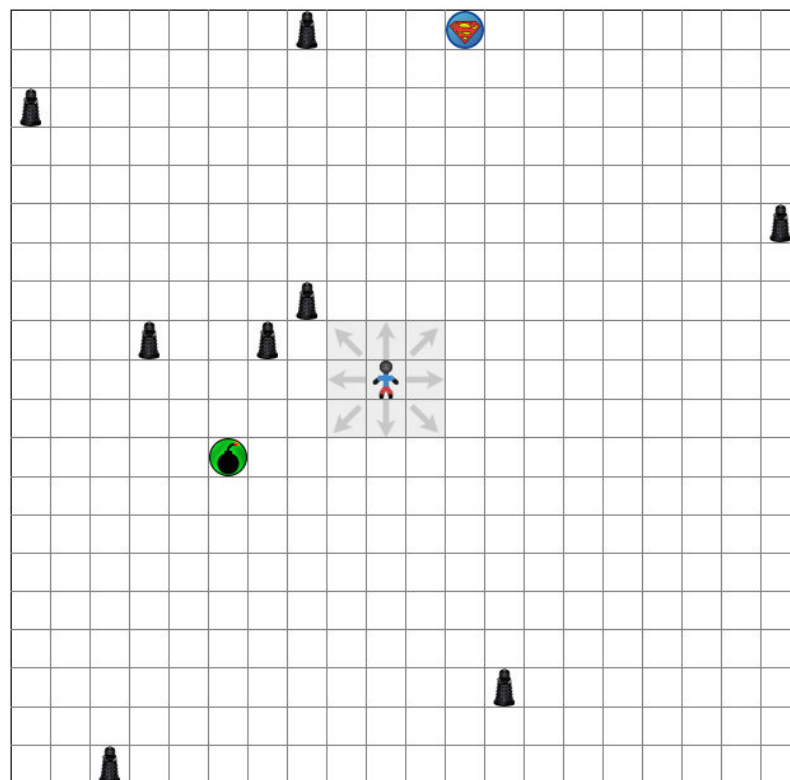
El juego Robots.

En esta adaptación del juego interactúan un grupo de robots y un jugador (humano), que debe escapar y defenderse de la banda de robots que sin ninguna razón explicable, intentan matarlo.

Es un juego de turnos, es decir los movimientos tanto del jugador como de los robots se dan un turno a la vez. El jugador puede disparar (y destruir) a los robots, además que aprovechan una debilidad de los robots para destruirlos: cuando dos robots chocan entre sí, se autodestruyen.

El juego se desarrolla sobre una matriz de casillas en donde en cada posición puede ubicarse nuestro jugador o un robot. El tamaño de la matriz debe ser 13 x 13. Los movimientos se hacen una casilla a la vez, en cualquiera de las 8 direcciones posibles, en caso que se sobrepase el límite del tablero (matriz), se debe hacer una rotación, es decir, el jugador va a aparecer al otro extremo de la matriz.

Después de cada movimiento del jugador, todos los robots se mueven una casilla en una dirección aleatoria. Al inicio del juego, los robots deben distribuirse aleatoriamente en el tablero, y el jugador se ubica en el centro de la matriz.



Teleport Bomb Restart Wait

0 ❤️ 1 🚒 1 🌱

Level: 1 Enemies remaining: 8 Score: 0 High score: 7579

La cantidad inicial de robots en nuestro caso se empieza con 10 robots. Cada vez que dos robots chocan (es decir, caen en la misma casilla), se produce una explosión y un incendio que perdura durante el resto del juego. Si otro robot o el jugador pasan sobre uno de estos incendios posteriormente, muere.

Cuando un robot alcanza al jugador, este muere y el juego termina. Cuando el jugador logra destruir a todos los robots, pasa a una siguiente etapa con más robots (cambio de nivel). El incremento en la cantidad de robots es de 5 por cada nivel.

Debe llevarse un control de puntos para el jugador, que inicia en 0. Por cada movimiento del jugador se suman 5 puntos al jugador.

El jugador cuenta con una cantidad limitada de disparos, que inicia con n disparos. Cada vez que hay cambio de nivel se incrementa en m la cantidad de disparos. Estos datos de cantidad inicial de disparos e incremento de disparos por nivel, son argumentos de configuración del juego.

Debe definirse algún mecanismo que permita disparar en alguna de las 8 direcciones posibles. Para que un disparo sea efectivo, el robot deberá estar a 2 o menos casillas de distancia. En caso que se dispare a más de 2 casillas, el disparo se pierde.

El jugador cuenta con bombas, que inicia con una cantidad de a bombas. Cada vez que hay cambio de nivel, se aumenta en 1 la cantidad de bombas. La cantidad inicial de bombas es un argumento de configuración del juego. Las bombas son un mecanismo de defensa del jugador, pueden ser ubicadas en cualquier parte del tablero. En caso que un robot llegue a una casilla con bomba, el robot muere y la casilla vuelve a quedar disponible.

El jugador cuenta con una cantidad de teletransportaciones aleatorias para escapar cuando se encuentra acorralado. Esto produce que el jugador sea ubicado en una casilla aleatoria del tablero. Si el jugador cae sobre un robot o sobre un incendio, muere instantáneamente y el juego termina. Cada teletransportación aleatoria disminuye 5 puntos al jugador, debiendo verificar si tiene puntos suficientes para realizarla. La cantidad de teletransportaciones aleatorias es un argumento de configuración.

Además, el jugador también cuenta con una cantidad limitada de b teletransportaciones seguras. Estos movimientos también son aleatorios, pero garantizan que en la casilla que el jugador cae no haya robots cerca (a una casilla de distancia) y que no exista ahí un incendio. Este tipo de transportación resta 25 puntos al jugador. Cada vez que hay un cambio de nivel se incrementa en uno la cantidad de teletransportaciones seguras.

Configuración.

Permite definir todos los valores necesarios para el funcionamiento del juego.

Ayuda interactiva.

Permite obtener ayuda sobre la utilización del juego.

Salón de la fama.

Permite almacenar en un archivo de texto, el nombre y puntaje de los 3 mejores registros de juego. Solamente se van a mantener los 3 mejores registros, debe existir una opción que permita visualizar los datos de los jugadores que están en este salón de la fama.

3. Funciones a investigar.

Para realizar la tarea, debe **investigarse** el uso de algunos algoritmos y funciones. La investigación a realizar debe abarcar:

Manejo interfaz gráfica

Utilización de multimedia: integración de animaciones, sonidos y otros.

Manejo de archivos de texto.

4. Documentación.

La documentación es un aspecto de gran importancia en la elaboración de programas. Existen 2 tipos de documentación a realizar para esta tarea: la interna y la externa.

La documentación interna se refiere a la inclusión de comentarios en el programa fuente, al menos antes de definir cada función, y en los cuales se explique lo que realiza la función, las entradas, salidas y restricciones consideradas. En las funciones que definen la interfase gráfica únicamente se debe identificar el componente utilizado.

La documentación externa debe incluir:

- Tabla de contenidos
- Introducción
- Descripción del problema.
- Análisis de resultados. (incluyendo corridas de ejemplo)
- Dificultades encontradas: problemas en el desarrollo y que se hizo para corregirlos
- Bitácora de actividades: se deben ir anotando todas las actividades, tipo de actividad, su descripción y duración.
- Estadística de tiempos: un cuadro que muestre un resumen de la Bitácora de Actividades en cuanto las horas **REALES** invertidas. Ejemplo:

FUNCION	Tiempo
Análisis de requerimientos	xx horas
Diseño de la aplicación	xx horas
Investigación de funciones	xx horas
Programación	xx horas
Documentación interna	xx horas
Pruebas	xx horas
Elaboración documento	xx horas
TOTAL	xx horas

- Conclusión personal del proyecto.

5. Evaluación.

Documentación		20%
Interna	5%	
Externa	15%	
Resultados (ejecución, eficiencia, estructuras planteadas, presentación)		
Funciones:		
Manejo de la interfaz		20%
Juego		40%
Configuración		5%
Ayuda		5%
Salón de la fama		10%

6. Aspectos Administrativos.

- La tarea es **individual** y se debe entregar hasta el día 16/10/2014 hasta las 11:59 pm, en forma electrónica, en un archivo comprimido con el nombre del estudiante, que contenga TODO lo necesario para poder ejecutarla. En la clase siguiente se asignará una cita de revisión del programa. **No se aceptarán tareas después de la fecha y hora indicadas.** Debe enviarse un archivo readme.txt con la versión de Python a utilizar para la revisión y alguna otra indicación que se considere importante. Se debe enviar al correo danielcanessa09@gmail.com
- La defensa o revisión del proyecto es indispensable. En esta revisión se preguntará sobre aspectos relacionados con funcionalidad, así como sobre el código. El estudiante debe mostrar TOTAL dominio de estos dos temas, de lo contrario, el proyecto puede ser considerado como una copia.
- En caso de probarse algún tipo de fraude en la elaboración de la tarea se aplicarán todas las medidas indicadas al inicio del curso, incluyendo una carta al expediente del estudiante.
- Se debe incluir en el archivo comprimido la documentación solicitada. Debe entregarse en formato electrónico (archivo .doc .odt o .pdf).
- No se aceptarán tareas cuyo archivo sobrepase 2 mb de espacio en disco.
- Cualquier falta a los aspectos aquí enunciados implicará pérdida de puntos.
- **El profesor se reserva el derecho de calificar forma y fondo de las actividades tomando como referencia la mejor actividad presentada**

7. Bibliografía.

Documentación técnica Python