

Examen final: *Juego en LAN*

Universidad de San Carlos de Guatemala
Facultad de Ingeniería
EIME - IPC1 (0769) "N"

Ing. Iván René Morales
1º Semestre 2014

¿Qué debe hacer?

Trabajando en pareja, o de forma individual, con Python 2.7 implementar un juego en línea similar a *Batalla Naval*.

Se jugará entre dos computadoras (o una sola computadora en la dirección de *loopback* 127.0.0.1) a través de TCP. Un jugador será el servidor, y el otro el cliente.

Cada usuario deberá tener una interfaz idéntica, pero el servidor (inicialmente) quedará a la espera de la conexión proveniente del cliente. Será el cliente quien inicie a jugar, y los turnos se irán alternando cada vez que un jugador realice un disparo.

¿Cómo jugar?

Inicialmente, deberán mostrarse los mapas (principal y secundario), incluyendo a los barcos que el usuario ha colocado previamente en el archivo de texto *ExamenFinal.txt*. Para más detalles sobre cómo desarrollar el juego, diríjase la sección *Documentos de ayuda*.

Restricciones y Limitaciones

Los barcos disponibles no deben ser específicamente los que establecen las reglas oficiales (*portaaviones*, *submarino*, *destructor*, *buque*, etc.): uno podrá elegir el tamaño de cada barco a discreción (siempre que cumplan con los límites del mapa y no exista traslape entre ellos). Los barcos deben estar solamente en posición horizontal.

Se deben imprimir dos mapas (principal y secundario). En el mapa principal se mostrarán los barcos ubicados inicialmente, así como los disparos recibidos por el adversario. En el mapa secundario se desplegarán los disparos fallidos

y acertados hacia el oponente. En ambos casos (mapa principal y secundario) deben mostrarse los números correspondientes a cada fila y columna. Los mapas de ambos jugadores deben ser del mismo tamaño: es necesario que este valor resida en una constante/variable, con el fin de cambiar las dimensiones del juego al hacerle un cambio.

Queda a discreción de los/las programadores(as) la simbología a utilizar para cada barco, disparo y límite del mapa.

Deberá realizar validación de datos a todo texto que el usuario ingrese a través del teclado.

Un archivo de texto llamado *ExamenFinal.txt* será creado por el usuario antes de ejecutar el *script*. Aquí se encontrará la información del tamaño de cada barco y la ubicación inicial de éstos. El formato del archivo es de libre elección.

Al cargar el archivo de texto, antes de incluir las naves al mapa, se deberá verificar que todos los barcos queden dentro del perímetro del establecido, y que ningún barco se traslape con otro. Si alguno de los problemas anteriores se presenta, deberá mostrarse al usuario una **advertencia**, indicando qué barcos están en conflicto, y que serán descartados. Luego de pasar por el proceso de depuración, los barcos pueden agregarse como objetos al mapa principal.

El usuario no debe ser capaz de cambiar las dimensiones del tablero, ni la cantidad máxima de barcos (o espacios ocupados por barcos) durante la ejecución del juego. Aunque esto no debe impedir que durante la calificación, al modificar una constante/variable correspondiente a cada parámetro, las reglas del juego sean modificadas. Los valores por defecto para estos parámetros deben ser:

- **Máxima cantidad de barcos:** 4
- **Máximos espacios ocupados:** 15
- **Tamaño utilizable del mapa:** 10 x 10

Debe mostrar a cada jugador el instante en el que éste ha perdido o ganado a través de un mensaje (la ejecución del programa no debe fallar en este instante).

Cada vez que se imprima el mapa principal, deberá mostrarse la ubicación de los barcos que siguen vivos aún. Asimismo, si un barco es **completamente** destruido por el adversario, se mostrará un mensaje indicándolo: este barco ya no tiene que figurar en el listado de barcos en juego.

Cuando un jugador realiza un disparo hacia el contrincante, deberá indicársele que el oponente está planeando su ataque, y que debe esperar a que el usuario en la otra computadora juegue su turno.

Los únicos módulos que podrán importarse a Python serán: *socket* y *time*. La conexión LAN (ya sea a través de Ethernet, WiFi o local) debe realizarse con un socket TCP.

Existe una plantilla (*ExamenFinalSk.py*) que deberá descargar para comenzar a trabajar. Allí se describe la funcionalidad de cada clase, y el orden en el que se debe implementar cada una de estas. El link para descargar el archivo lo encuentra en la sección *Documentos de ayuda*.

Documentos de ayuda

- Reglas y metodología del juego:
([http://en.wikipedia.org/wiki/Battleship_\(game\)](http://en.wikipedia.org/wiki/Battleship_(game)))
- Video en *YouTube* con un ejemplo del funcionamiento:
<http://youtu.be/yby0Kf5B8wU>
- Código fuente base para iniciar el proyecto (plantilla):
<http://goo.gl/ybPCWG>

Premiación

A los mejores tres proyectos se les retribuirá su trabajo con un premio (aparte de la satisfacción propia de cada estudiante y una buena nota) que será de utilidad en el desarrollo de algunos proyectos de su carrera. Los premios consisten en *hardware* que podrá conectar en forma de sensor o interfaz visual a microcontroladores. Si desean comprar su kit e iniciar de una vez en el mundo de hardware programable (esto no es propaganda), el MSP430 Launchpad (\$9.99) es una buena opción: <http://goo.gl/pFi9cC>

Deberán subir un video a *YouTube* (no más de **3 minutos**) donde muestren de forma creativa su proyecto (hay un espacio en el formulario de entrega para que coloquen allí el link). Alguna funcionalidad adicional a la requerida sería un extra que podría darle más mérito al ser calificado por el jurado.

Miembros del jurado

El jurado será conformado por las personas **asignadas** oficialmente al curso, los auxiliares y el profesor. Luego de pasada la fecha de entrega del proyecto, se les enviará por correo electrónico a los estudiantes los videos de sus compañeros, así como la página donde podrán realizar la votación. Como regla, los alumnos no pueden votar por el proyecto que ellos(as) mismos(as) realizaron, ni votar más de una vez.

Los premios

Primer Lugar

Puede elegir entre:

- 1 Módulo Bluetooth HC-06:
<http://goo.gl/TuV8AZ>
- 1 Sensor Ultrasónico de distancia HC-SR04 para cada integrante del grupo:
<http://goo.gl/JdB0i5>

Segundo Lugar

1 Acelerómetro analógico de 3 ejes MMA7361 para cada integrante del grupo:
<http://goo.gl/9QQRgA>

Tercer Lugar

1 Display de 4 caracteres de 7 segmentos para cada integrante del grupo:
<http://goo.gl/bIY0HD>

Metodología de entrega

Con el fin de agilizar el proceso de calificación y evitar hacinamiento de los alumnos, la entrega será a través de un formulario en la web, pero la calificación se realizará en distintos horarios. Esta calendarización será publicada luego de la fecha de entrega.

El proyecto se trabaja en parejas o individualmente, y cualquier indicio de copia (parcial o total) con otro grupo será motivo de anulación inmediata. El límite de la entrega del proyecto es el sábado 17/05/2014 a las 18:00 hrs. Este proceso será via web: <http://goo.gl/hNMpBY>. Puede enviar su examen solamente una vez, de lo contrario no será calificado. *Todo debe estar debidamente indentado y comentado.* Al momento de la calificación deberá lleva 2 COPIAS IMPRESAS de la última página de este documento. No es necesario que lleve computadora en este instante.

Calificación

Carnet 1: _____ Carnet 2: _____

Descripción	Calificación
Uso adecuado de clases	/30
Uso de socket TCP	/20
Implementación de métodos	/15
Restricciones de juego	/15
Interfaz del usuario	/10
Manejo de archivo de texto	/10

Nota:

Calificó: