Enunciado Trabajo Práctico

Laberinto

Redes de Datos

Ing. Diego Azcurra – Ing. Damián Santos – Lic. Gustavo Fernandez

UNLA

2017

|  |  |  |
| --- | --- | --- |
| DNI | Nombre | Evaluación Individual |
| 34.382.314 | Euen Lopez |  |
| 36.086.446 | Javier Vescio |  |
| 94.557.769 | Jose Melgarejo |  |
| 38.125.121 | Agustín Padrón |  |
|  |  |  |
| Evaluación Trabajo |  | |

## Objetivo

Desarrollar una aplicación que permita jugar en un laberinto bajo el esquema de cliente y servidor.

## Descripción

## Laberinto

El laberinto es un área que cuenta con una única entrada y una única salida. Desde esa entrada surgen varios caminos de los cuales al menos uno permite llegar a la salida. Para salir, primero es necesario encontrar una llave. A su vez, a lo largo del laberinto se podrá encontrar oro. El oro sirve para “pagar” a los guardias que vigilan los caminos. Si nos cruzamos con un guardia y no tenemos oro, nos matará y tendremos que empezar de nuevo.

El laberinto es una matriz cuadrada de una determinada dimensión X. En cada posición, se podrá encontrar un carácter que especificará:

* E: Entrada.
* P: Pared.
* C: Camino.
* K: Llave.
* S: Salida.
* O: Oro.
* G: Guardia.

## Objetivo

El objetivo del juego es encontrar la salida recolectando la mayor cantidad de oro posible.

## Arquitectura

El juego se desarrollará bajo una arquitectura cliente servidor. La aplicación cliente será el jugador y se conectará al servidor para solicitar todos los elementos del juego.

## Desarrollo del juego

Para poder jugar, el cliente deberá conectarse al servidor mediante un usuario y contraseña.

El servidor posiciona al jugador en la casilla indicada con la letra E. En todos los casos, el servidor va revelar solo partes del laberinto: matrices cuadradas armadas desde un parámetro Y (desde la ubicación del jugador, se arma un cuadrado avanzando Y casillas en todos los sentidos). Desde esa posición, el jugador podrá moverse siguiendo los casilleros marcados con C. Al posicionarse en la nueva ubicación, solicitará al servidor que revele las casillas correspondientes.

Por ejemplo, tomemos una parte de un determinado laberinto:

000000000111111

123456789012345

01|PPPPPPPPPPPPPPP

02|ECCCCPPPCCCCCCP

03|PPPPCPPPCPPPPPP

04|PPPPCCCCCPPPPPP

05|PPPPCPPPCCCCCCP

06|PPPPCPCCCPPCPPP

07|PPPCCPPPCCCCPPP

El juego comienza en la posición (2,1). Suponiendo Y= 2, el servidor en este caso va a revelar únicamente:

000

123

01|PPP

02|ECC

03|PPP

04|PPP

Desde ahí, en este caso sólo podrá moverse en sentido horizontal a la posición (2,2). Para esto, el cliente enviará un comando de movimiento a la derecha y el servidor responderá con la nueva matriz:

0000

1234

01|PPPP

02|ECCC

03|PPPP

04|PPPP

La secuencia continúa en búsqueda de oro, de la llave y de la salida del laberinto.

## Requerimientos

Se pide desarrollar un sistema cliente/servidor que permita:

1. Servidor:
   * El servidor deberá permitir loggear jugadores.
   * Deberá administrar el juego. Recibirá comandos desde los jugadores y enviará respuestas.
2. Cliente:
   * Deberá poder conectarse a un servidor ingresando IP y puerto del mismo.
   * Luego de recibir la posición inicial del jugador, deberá interpretar los comandos ingresados por teclado, enviarlos al servidor y actuar en consecuencia.
3. Protocolo:
   * Todos los comandos que se intercambian entre cliente y servidor deberán respetar un protocolo definido por cada grupo. Como mínimo, se deberán contemplar los siguientes campos:
     + Encabezado:
       - Tipo de comando (Conexión, Movimiento, Tomar (para el oro o las llaves), Pagar, Abrir Puerta, etc).
       - Tamaño del mensaje: salvo que se decida trabajar con tamaños fijos.
     + Datos:
       - Datos de los distintos mensajes en el formato que decida el grupo.
   * El protocolo deberá ser documentado detalladamente. Tomar como referencia algún estándar de la industria (HTTP, FTP, telnet…). Por ejemplo:
     + - <https://es.wikipedia.org/wiki/Protocolo_de_transferencia_de_hipertexto>
       - <https://es.wikipedia.org/wiki/Anexo:C%C3%B3digos_de_estado_HTTP>
4. Seguridad:

El sistema deberá implementar al menos dos servicios de seguridad informática. El grupo seleccionará qué servicios y con qué implementaciones.

1. Consola cliente:

Se deberá desarrollar otra aplicación cliente que permita interactuar con el servidor mediante mensajes de pregunta y respuesta en formato texto (de la misma forma que lo hace el juego pero en forma manual).

El protocolo definido tendrá una cantidad de operaciones a las que se podrá acceder mediante un esquema de request/response. Esta aplicación deberá permitir que el usuario ingrese por consola un comando para que el cliente lo envíe al servidor y muestre la respuesta obtenida.

Por ejemplo, si el protocolo define que el mensaje de login es:

LOG|us:Damian|pass:Damian

Y la respuesta es:

LOG|200|ok

El usuario podrá ingresar esa cadena y deberá recibir la respuesta esperada. Si algún dato está mal, deberá recibir un código de error.

## Presentación

23/10/17: **Entrega del TP**.

06/11/17: **Recuperatorio del TP**.

**Condiciones de entrega**.

El trabajo entregado deberá contener:

* Una copia impresa del enunciado del trabajo práctico (TODO este documento, incluyendo los anexos).
* Una copia impresa de la estrategia de resolución del trabajo práctico. Es un texto descriptivo de cómo se estructuró la aplicación, como se separaron las capas, relaciones entre las entidades, es decir, todo aquello que consideren significativo para explicar la resolución del trabajo.
* El detalle del protocolo utilizado.
* El detalle de los servicios de seguridad informática aplicados y la forma en que fueron implementados.
* Una copia impresa de todos los archivos del proyecto. Poner como encabezado de cada hoja el nombre del archivo. Pueden imprimirse dos páginas por hoja, en formato horizontal.
* Un CD conteniendo en formato digital todos los puntos anteriores y el proyecto completo.
* La presentación deberá ser en un folio o carpeta, en forma prolija y debidamente identificada. Los CDs deberán contener el número de grupo y el nombre y los padrones o documentos de identidad de cada uno de los integrantes y deberán estar correctamente adjuntos al resto del trabajo práctico de forma tal que no puedan perderse. Además, deberá incluirse **todo** el proyecto desarrollado (**la carpeta completa** generada por el IDE, con los archivos del proyecto y el código fuente) incluyendo los archivos de pruebas.
* Todas las hojas deben estar numeradas.

El incumplimiento de cualquiera de las normas de entrega implicará la desaprobación del trabajo práctico.

Metodología de evaluación:

La Evaluación de los trabajos prácticos contará con una etapa grupal y una individual.

* Grupal: Se realizará un conjunto de pruebas sobre el trabajo presentado por los alumnos en presencia de los mismos. Se deberá aprobar la totalidad de las pruebas. En caso de que una prueba falle, los alumnos podrán intentar corregir el código mientras dure la evaluación.
* Individual: Se realizará una evaluación individual oral o escrita para cada alumno. Los temas a evaluar podrán ser, por ejemplo: preguntas teóricas sobre el contenido de la materia, preguntas sobre el trabajo práctico, codificación de alguna primitiva o modificación del trabajo práctico, etc.

La nota final del trabajo se calculará en función de las notas obtenidas en forma grupal e individual. La nota grupal será el promedio entre la primer presentación y el recuperatorio (en caso de necesitarlo). Por este motivo, SOLO deberán presentarse aquellos grupos que hayan concluido TODO el trabajo práctico ya que no se harán evaluaciones parciales.

## Revisiones

Se agregó el punto 5 dentro de los requerimientos.

ESTRATEGIA DE RESOLUCION DE TRABAJO PRACTICO

Mediante una arquitectura de cliente-servidor y utilizando sockets, hemos resuelto que toda la lógica del sistema sería implementada del lado del servidor.

El seguimiento de la posición del jugador a lo largo del laberinto, el análisis de la posibilidad de realizar o no un movimiento, la lógica que determina cuando el jugador pierde la partida, fue colocada en el lado del servidor.

El cliente solamente se limita a enviar peticiones al servidor (identificación, movimiento y salida) y en base a las respuestas de éste, va actualizando el estado de la aplicación.

Lo hicimos en Python, un servidor que escucha las solicitudes de conexión de los clientes; una vez que el servidor detecta que alguien se conecta, crea un hilo para atender a ese cliente, y a la vez, mantiene el hilo principal en la escucha de nuevas conexiones. En ese momento se espera que el usuario se autentique. Si la autenticación falla, el server cierra la conexión a ese cliente, enviando el mensaje de error correspondiente.

En caso de que la autenticación sea exitosa, el server aceptará los comandos referidos al movimiento en el laberinto y la salía del juego.

SEGURIDAD

Para una mayor seguridad, todos los paquetes que envía el cliente al servidor son encriptados.

Luego, el servidor desencripta el paquete, usando una clave maestra que fue definida en el cliente.

Las contraseñas recibidas son encriptadas en MD5 y se comparan con las almacenadas en la base de datos.

El paquete posee un encabezado de tres letras, que contiene el comando, seguido de los parámetros del mismo.

COMANDOS

**1.-LOGUEO**

**Solicitud**: log|usuario|clave:

**Respuesta**: log|ok (si se acepta el logueo); log|no (si autenticación falla)

**2.-MOVIMIENTO**

**Solicitud:** mov|sentidoMovimiento():´

**Respuesta**: map|minimapa()-oro()-llave()-msg(sentidoMovimiento);map|minimapa()-oro()-llave()-msg(“error”);

**Referencias:**

Minimapa() = string

Oro() = int

Llave() = boolean

Msg() = string

sentidoMovimiento() = “ar” (arriba),”ab” (abajo),”iz” (izquierda),”de” (derecha).

**3.-SALIDA**

**Solicitud:** exit|:

**Respuesta:** exit|ok

**4.-SOLICITUD INEXISTENTE**

**Solicitud:** textoInvalido|:

**Respuesta:** err|msg(“Comando inválido”)

Anexo – Correcciones

|  |  |  |
| --- | --- | --- |
| Prueba | Resultado | Comentario |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Anexo – Correcciones (copia para los alumnos)

|  |  |  |
| --- | --- | --- |
| Prueba | Resultado | Comentario |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |