

Trabajo Práctico N° 2

Los objetivos de este trabajo práctico son afianzar los conocimientos de programación orientada a objetos y aplicar las estructuras de datos y los patrones de diseño vistos en clase.

1. Pautas de Trabajo y Evaluación

1.1) Registro de los grupos

De dos a cuatro estudiantes. Por favor completar el formulario indicado abajo con los integrantes de cada grupo. Recuerden usar nombres nuevos para los nuevos grupos.

Formulario: <https://forms.gle/LzC3wCQfTrJnVPAe9>

1.2) Entregas

Están previstas dos entregas. En la primera, se deberá entregar un diseño preliminar de la solución. En la segunda, la solución deberá estar completa y operativa.

Entrega	Comisión	
	Martes y Jueves (mañana y tarde)	Miércoles y Viernes
Parcial (Diagrama de Actividades/Flujo y Diagrama de Clases y Objetos)	Martes 28/05/2019 (mañana) Jueves 30/05/2019 (tarde)	Viernes 31/05/2019
Completa	Martes 18/06/2019	Viernes 21/06/2019

1.3) Cómo entregar

El TP se deberá entregar en formato impreso (sólo el informe) y digital (tanto el informe como el código), hasta las 23:59:59 del plazo solicitado.

Se deberá enviar a algo-untref-rw-doc@googlegroups.com un archivo .zip con la carpeta del proyecto. El archivo deberá tener la forma:

tp2-grupo-<nombre_del_grupo>-turno-<mar_jue o mie_vie>.zip

Respetando mayúsculas y minúsculas.

(Ejemplo: tp2-grupo-nombre-turno-mar_jue.zip)

El tamaño máximo total del mensaje es de 8 MB.

1.4) Estructura

- Código Fuente: deberá cumplir las convenciones de Java
 - Código principal en /src/main/java/<namespace/clases>
 - Tests en src/test/java/<namespace/clases>
 - Recursos necesarios en /src/{main,test}/resources
- Además, deberán incluirse todos los Archivos necesarios para, desde Eclipse, poder importar el proyecto, compilarlo, ejecutarlo y probarlo (ej.: archivos .project, .classpath, etc.)
- También deberán adjuntar un Informe en un archivo README.md ubicado en el directorio raíz, el cual incluirá:
 - Nombres de los integrantes del grupo
 - Decisiones de diseño tomadas
 - Diagramas que ilustren dichas decisiones
 - Diagrama de Clases
 - Diagrama de Objetos
 - Otros que considere pertinentes
 - Descripción de cada archivo *.java comprendido en solución del problema
 - Conclusiones
- Aclaren todo lo que consideren importante, aún lo que consideren trivial.

1.5) Revisiones semanales

Cada grupo tiene un docente responsable de su seguimiento. Recomendamos que coordinen revisiones al menos, una vez por semana.

1.5.1) Metodología recomendada

Para agilizar el tiempo de revisión (lo ideal es llevarlo a 10'), se les solicita que preparen cada reunión.

Luego, por favor tengan preparadas las respuestas a la siguientes preguntas. (Para ir viendo su propio progreso, vayan agregando cada una de las respuestas en una tabla al README del proyecto.)

- ¿Qué hicieron desde la última revisión?
- ¿Qué problemas tuvieron? ¿Cómo los solucionaron?
- ¿Qué aprendieron?
- ¿Qué piensan hacer hasta la última revisión?

1.6) Evaluación

Si bien la idea es minimizar comentarios redundantes dentro del código, recuerden que

esta es una instancia de evaluación: aclaren todo lo que consideren importante, aún lo que consideren trivial.

1.6.1) Criterios

Se evalúa la Aplicación Práctica de Conceptos vistos en la asignatura.

1.6.1.1) Subjetivos

La Metodología de Trabajo.

Organización del Trabajo

¿Cómo se lo vio trabajar al grupo?

Dedicación y Esfuerzo

¿Fueron capaces de ir más allá de lo estrictamente necesario para aprobar?

1.6.1.2) Objetivos

Análisis y Diseño Orientado a Objetos

- Uso correcto de Herencia, Composición y Polimorfismo.
- Desarrollo Dirigido por Tests (Test-Driven Development).

Uso Eficiente de Estructuras de Datos

Analizar las fortalezas y debilidades de cada una de las EDs vistas en clase (y de sus distintas implementaciones posibles), y elegir la combinación que resuelva cada componente del problema bajo consideración con la máxima eficiencia posible.

Uso Correcto de Recursividad

Uso de Patrones de Diseño

Aspectos básicos de Calidad Interna del Código

- Uso correcto de las Estructuras de Control provistas por el lenguaje
- Organización y Modularidad; Cohesión y Acoplamiento

2. Ejercicio: PacSim

2.1. Planteo

Un juego de PacSim consta de un tablero de $M \times N$ con una entrada y una salida marcadas. En el tablero hay casilleros marcados con paredes por los que no se puede pasar. Una posición en el casillero marca el lugar donde nuestro personaje (Pac) está parado.

Al inicio del juego Pac nace con una cantidad de vidas y parado en la entrada, sin provisiones.

En algunos casilleros del camino hay minas y en algunos otros hay provisiones. Hay a lo sumo una mina en un casillero.

Las provisiones pueden ser:

- escudo de fuerza
- caja de vitaminas, y
- más provisiones.

Una caja de vitaminas le permite a Pac ganar una vida.

Un escudo de fuerza tiene una cantidad fija de puntos de fuerza cuando Pac lo encuentra, y le permite entrar a un casillero que contiene una mina, destruir la mina y no perder una vida. El escudo de fuerza pierde un punto de fuerza cada vez que Pac lo usa para destruir una mina.

Si Pac entra sin escudo a un casillero que contiene una mina, la mina explota y Pac pierde una vida.

Las movidas posibles para Pac (si todavía tiene vidas) están dadas por:

- ir arriba
- ir abajo
- ir a la izquierda, o
- ir a la derecha

Siempre y cuando no se salga del tablero ni se pise una pared.

Al llegar a la salida, Pac gana; el puntaje ganado es igual a la cantidad de vidas que le quedaron.

Si se queda sin vidas, Pac pierde el partido.

2.2. Interfaz de usuario

La interfaz con el usuario será a través de la consola.

Al iniciar, el sistema deberá leer un archivo de configuración donde se especifican el tamaño del tablero, la ubicación de las paredes, la ubicación de las bombas y la ubicación de las provisiones. Al iniciar deberá presentar el jugador

1. La cantidad de vida y provisiones de las que dispone
2. El tablero y la posición del jugador, con las paredes identificadas.
3. Luego le presenta al jugador las opciones de movimiento:
 - a. arriba
 - b. abajo
 - c. derecha
 - d. izquierda

Cuando el jugador se mueve a una nueva celda, el sistema calcula los daños que sufre el jugador si es que habían bombas, actualiza las defensas disponibles si encontró un escudo y actualiza la cantidad de vidas si encontró una caja de vitamina y vuelve al punto 1

Esta mecánica se repite hasta que el jugador llega a la celda de salida o hasta que no le quedan vidas para seguir avanzando.

En ambos casos muestra el puntaje final obtenido y propone iniciar una partida.

Ejemplo de tablero inicial con el jugador en la posición de entrada

Referencias:

- S: Salida
- J: Jugador
- X: Pared

```

      | 1 | 2 | 3 | 4 | 5 | 6 | 7 | |
|1 | X | X | X | X | X | X | X |
|2 |           | S |
|3 |           | X | X |
|4 |           | X |
|5 | X | X | X | X | X | X |
|6 |           | X |
|7 | X | X | X |   | X | X |
|8 | X | X | X |   | X | X | X |
|9 | X |           | X |
|10| X | J | X | X | X | X | X |
    
```

2.3. Archivo de configuración

El archivo de configuración cuenta con 4 secciones:

[TABLERO] [MINAS] [PROVISIONES] [UBICACION_PROVISIONES]

En la sección [Tablero] se definen las siguientes variables:

M: cantidad de columnas

N: cantidad de filas

S: ubicación de la salida

E: ubicación de la entrada

L: cadena de 0 y 1 donde 1 indica pared y 0 que no hay pared en esa posición del laberinto

En la sección [MINAS] se define una única variable:

B: lista de ubicaciones de minas, separadas por coma

En la sección [PROVISIONES] se definen las provisiones disponibles con el siguiente formato: cada provisión se define en una línea con la letra P seguida de un número para identificar dicha provisión, un signo igual y una lista de valores separados por coma. Los valores pueden ser V: vitamina, E: escudo, Pn: referencia a otra provisión ya definida

En la sección [UBICACION_PROVISIONES] se definen las posiciones donde se encuentran las provisiones con el formato Pn= lista de ubicaciones.

Por ejemplo dado el siguiente tablero:

0	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	32	33	34
35	36	37	38	39	40	41
42	43	44	45	46	47	48
49	50	51	52	53	54	55
56	57	58	59	60	61	62
63	64	65	66	67	68	69

	PARED
	MINA
	PROVISION
	MINA Y PROVISIÓN

El archivo de configuración correspondiente sería:

[TABLERO]

M=7

N=10

S=13

E=64

L=111111100000000000001100000011111010000001111001111101111000000110
111111

[MINAS]

B=11,16,17,18,42,43,44,51,56,57,58

[PROVISIONES]

P1=E

P2=V

P3=E,E

P4=P1,P1,P2

P5=P4,P1

P6=P5,V,P3

[UBICACION_PROVISIONES]

P1=7

P2=14

P3=21

P4=43

P5=56

P6=64,42