

Inmate

PROGRAMACIÓN 2018-2019

GRUPO Nº 13

“El buen código es su mejor documentación.” - Steve McConnell

Tabla de contenido

--	--

Universidad Politécnica De Madrid

Tabla de contenido

--	--

Tabla de contenido

Tabla de contenido

TEMA	1
¿QUÉ ES?	1
¿PARA QUÉ SIRVE?	1
FUNCIONALIDADES	2
INTRODUCCIÓN DE COMANDOS	2
COGER OBJETOS	2
INVENTARIO	2
CAMBIO NIVEL DIFICULTAD	2
GUARDAR PISTAS ENCONTRADAS	2
DOS POSIBLES FINALES	2
SALIR DEL JUEGO EN CUALQUIER MOMENTO	3
MINIJUEGO: TRES EN RAYA	3
MINIJUEGO: HANGMAN/AHORCADO	3
CÓDIGO FINAL ALEATORIO	3
DEPENDENCIA/CONDICIONES DURANTE EL JUEGO	3
LECTURA DEL INVENTARIO	3
FUNCIONES	4
TRES EN RAYA(SARA)	4
HANGMAN(AIDA)	4
PIEDRA PAPEL TIJERA(FERNANDO)	4
UTENSILIOS	4
SALAS	9
DIAGRAMA DEL JUEGO	11
CONCLUSIÓN CRÍTICA	12
AIDA MUÑOZ	12
SARA BEATRIZ ALONSO	12
FERNANDO BELLIDO	12
COMO GRUPO:	12
CONCLUSIÓN	13
AIDA MUÑOZ	13
SARA BEATRIZ ALONSO	13
FERNANDO BELLIDO	13

Tabla de contenido



COMO GRUPO:	13
WEBGRAFÍA	14
REFERENCIAS	15

Tabla de contenido

"Pero, ¿para qué puede valer eso?"

- Ingeniero en la división de sistemas informáticos avanzados de IBM, hablando sobre los microchips, en 1968.

Tema

¿Qué es?

El proyecto *Inmate* se categoriza como "Text Adventure", o en español "Aventura Conversacional".

Una "Aventura Conversacional" es una simulación de un entorno, en el cual el usuario o jugador tendrá que resolver ciertos acertijos (o juegos) para poder finalizarlo. Sin embargo, todo esto se realiza sin ninguna interfaz gráfica, a través del uso de comandos y del texto que el programa devuelve como respuesta a dichos comandos.

¿Para qué sirve?

La utilidad fundamental de este proyecto, debido a su naturaleza, es simplemente el entretenimiento o pasa tiempos.

Funcionalidades

Introducción de comandos

La funcionalidad principal de este proyecto, es la posibilidad de insertar comandos. Cada sala tiene unos comandos predeterminados como son el movimiento entre salas, o la posibilidad de acceder a distintas líneas de la historia, además de comandos propios de cada sala.

Coger objetos

El juego tiene objetos en las diferentes salas, que pueden ser recogidos por el usuario a su paso por estas. Esta funcionalidad, como su nombre indica, permitirá que se puedan recoger dichos objetos encontrados.

Inventario

A raíz de la anterior funcionalidad, surge la necesidad de “guardar” los objetos recogidos, para poder acceder a ellos más adelante en el juego. Esta funcionalidad es la de crear un inventario accesible que recoja todos los objetos disponibles.

Cambio entre objetos

“Sub-funcionalidad” que permite cambiar objetos (intercambio de un objeto por otro).

Cambio nivel dificultad

Cualquier juego, por simple o complicado que sea, tiene un nivel de dificultad elegido por el jugador al empezar el juego, dicha elección hará que se vea afectado el juego.

Guardar pistas encontradas

La ambientación de este juego es un “Escape Room”, y algo característico de este tipo de juegos de aventura es encontrar pistas y usarlas a lo largo del juego para poder habilitar ciertas acciones.

Dos posibles finales

A lo largo del juego, el jugador se verá obligado a tomar ciertas decisiones que (inconscientemente) influirán en el capítulo final del juego. Es decir, algunas de las decisiones que deben tomar el jugador llevan a la muerte del personaje, y el resto permiten continuar jugando hasta que consigue escapar.

Tabla de contenido

Salir del juego en cualquier momento

Una funcionalidad esencial de cualquier juego es la posibilidad de salir (sin presionar el botón de cerrar el juego). Esta funcionalidad está disponible a lo largo de todo el juego

Nota: no se guardará el progreso del juego.

Minijuego: Tres en Raya

Funcionalidad que permite jugar al tres en raya. Influye en el final del juego

Minijuego: Hangman/Ahorcado

Funcionalidad que permite jugar al ahorcado. Influye en el final del juego

Minijuego: Descifrar código

“Sub-funcionalidad” que permite descifrar un código al final del juego, deducible por las pistas recibidas tras los minijuegos.

Aplicación del Ahorcado en el juego final (a través de números y no letras)

Minijuego: Piedra, Papel o Tijera

Funcionalidad que permite jugar al piedra, papel o tijera. Influye en el final del juego.

Código final aleatorio

A diferencia de otros juegos, que una vez jugados se conoce el código final, en Inmate se abre la posibilidad de que cada vez que se juegue, cambie el código final.

Dependencia/Condiciones durante el juego

Para poder avanzar en el juego, se requerirán ciertas condiciones/objetos que se irán consiguiendo a lo largo del juego.

Ej.: sí para poder realizar una acción se necesita un destornillador, dicha acción no se podrá realizar hasta y solo hasta que el jugador consiga el destornillador.

Lectura del inventario

El jugador podrá consultar su inventario siempre que este lo necesite.

Tabla de contenido

--	--

Funciones

Tres en Raya (Sara)

public static int [] marcadores (int[][]tablero)

Multiplicaciones del tres en raya

Parámetros: Tablero (Matriz Tridimensional)

Variables: marcadores

Condicionales: No

Bucles: No

Funcionalidades: Tres en Raya

public static int orientacionGanar (int[][] tablero)

Algoritmo del tres en Raya que evalúa si ganara o no

Parámetros: Tablero (Matriz Tridimensional)

Variables: n, res (Tipo Entero)

Condicionales: Sí, tipo if (3)

Bucles: Sí, (While)

Funcionalidades: Tres en Raya

public static int ganarFila (int [][] tablero)

Ganara la fila o no el programa, evaluación

Parámetros: Tablero (Matriz Tridimensional)

Variables: pos(Tipo entero)

Condicionales: Sí, tipo if (3)

Bucles: Si, (While)

Funcionalidades: Tres en Raya

Tabla de contenido

public static int ganarColumna (int[][] tablero)

Ganará la Columna o no el programa, evaluación

Parámetros: Tablero (Matriz Tridimensional)

Variables: fila, b, columna (Tipo entero)

Condicionales: Sí, tipo if (3)

Bucles: No

Funcionalidades: Tres en Raya

public static int orientacionPerder (int [][] tablero)

Similar a orientacionGanar solo que, al contrario

Parámetros: Tablero (Matriz Tridimensional)

Variables: n,res(Tipo Entero)

Condicionales: Sí, tipo if (3)

Bucles: Sí, (While)

Funcionalidades: Tres en Raya

public static int errorFila (int [][] tablero)

Estudio del tablero

Parámetros: Tablero (Matriz Tridimensional)

Variables: n, fila (Tipo Entero)

Condicionales: Sí, tipo if (4)

Bucles: Sí, (While)

Funcionalidades: Tres en Raya

public static int errorColumna (int[][] tablero)

Estudio del tablero por columna

Parámetros: Tablero (Matriz Tridimensional)

Tabla de contenido

Variables: n, a, columna (Tipo Entero)

Condicionales: Sí, tipo if (4)

Bucles: Sí, (While)

Funcionalidades: Tres en Raya

public static int tictactoe ()

El cuerpo del tictac toe

Parámetros: No

Variables: res, numeroMovimientos, filaMovimiento y ColumnaMovimiento
(Tipo Entero) ,tablero(Array Multidimensional)

Condicionales: Sí tipo if (4)

Bucles: Sí, (Do While)

Funcionalidades: Tres en Raya

Hangman (Aída)

public static int evaluarHangman (String palabra, char intento, int aPartirDe)

Evalúa el Ahorcado

Parámetros: palabra (String), Intento (Carácter), aPartieDe (Entero)

Variables: pos (Tipo entero)

Condicionales: Sí, tipo if (1)

Bucles: Sí, (For loop)

Funcionalidades: Hangman

public static boolean hangman(int dificultad)

Ahorcado

Parámetros: dificultad (Tipo entero)

Tabla de contenido

Variables: random, longitud, pos, fallosRestantes (Tipo entero), listaPalabras (Array tipo Strings), palabra, letrasAdivinadas (Tipo String), intento (Tipo Carácter), winner(boolean)

Condicionales: Sí, tipo if (4)

Bucles: Sí, (For loop, While)

Funcionalidades: Hangman

Aplicación Juego Final(Endgame)(Fernando):

public static void engame(int[] ingame)

Código final (Similar al ahorcado)

Parámetros: ingame (Tipo array de enteros)

Variables: code, numpad, porgress (Array 2 enteros y String) x (Input Int)

Condicionales: Sí, tipo if (2)

Bucles: Sí, (2 → 1 While y 1 For)

Funcionalidades: Sub-funcionalidad Hangman

Piedra, Papel o Tijera (Fernando)

public static String mov(int a)

Traduce el numero introducido a un valor (Piedra, papel o tijera)

Parámetros: a (Tipo entero)

Variables: String res

Condicionales: Sí, tipo Switch (1)

Bucles: No

Funcionalidades: Piedra, papel o tijera

Tabla de contenido

public static boolean checkers2(int x,int y)

Comprueba si la maquina ha ganado

Parámetros: x, y (Tipo entero)

Variables: who (tipo Boolean)

Condicionales: Sí, tipo if (1)

Bucles: No

Funcionalidades: Piedra, papel o tijera

public static boolean ppt1(int[]ingame)

El juego de piedra papel tijera, que tras ganar/perder te devuelve un boolean (True/False) según hayas ganado/perdido (Usado para devolver código)

Parámetros: Array (Tipo Entero)

Variables: scoreh, scorem, textoG, textoP, y, x(Input), troll (Tipo enteros todos)

Condicionales: Sí, tipo if (7)

Bucles: Sí (3) (Tipo DoWhile, y While x2)

Funcionalidades: Piedra, papel o tijera

Utensilios

Genéricos:

public static String printAr(int[] arr)

Devuelve un String formado por los elementos de un Array de String.

Parámetros: Array (Tipo Entero)

Variables: res (Tipo String)

Tabla de contenido

Condicionales: Sí tipo if(1)

Bucles: Sí, tipo for(1)

Funcionalidades: Inventario

public static String printAr2(String[] arr)

Devuelve un String formado por los elementos Array de Int.

Parámetros: Array (Tipo Entero)

Variables: res (Tipo String)

Condicionales: Sí, tipo if(1)

Bucles: Sí, tipo for(1)

Funcionalidades: Inventario y Sub funcionalidad Inventario, Lectura del inventario

public static int sumAr(int[] arr)

Devuelve la suma de los elementos de un array.

Parámetros: Array (Tipo Entero)

Variables: res (Tipo entero)

Condicionales: No

Bucles: Sí, tipo for(1)

Funcionalidades: Sub funcionalidad Inventario

Aplicados al Juego

public static int[] digits(int random)

Devuelve Array con los dígitos de un numero(random).

Parámetros: random(Tipo entero)

Variables: digits(Tipo Array Entero)

Condicionales: No

Tabla de contenido

Bucles: No

Funcionalidades: Código final aleatorio

public static int difficult()

Función que genera el número Inmate(Función Math.Random)

Parámetros: No

Variables: No

Condicionales: No

Bucles: No

Funcionalidades: Código final aleatorio

public static String objetoInventario (int a)

Función que devuelve el Objeto dado un numero entero

Parámetros: a(Tipo Entero)

Variables: res(Tipo String) y b(Tipo entero)

Condicionales: Si, (Tipo Switch)

Bucles: No

Funcionalidades: Lectura del inventario

public static void cambObjt(int[] inventory,int posobjt)

Función que imprime por pantalla el cambio de objetos realizado

Parámetros: posobjt (Tipo Entero) y inventory(Tipo Array)

Variables: objCogido/Dejado(Vienen de la funcion anterior), x(Input) e y (Ambos Enteros)

Condicionales: sí (2)

Bucles: No

Funcionalidades: Sub Funcionalidad Intercambio Objetos

Tabla de contenido

public static void elegObjt(int[] inventory)

Función que imprime por pantalla los objetos que se tienen en el inventario

Parámetros: inventory(Tipo Array)

Variables: res(Tipo String)

Condicionales: sí (2)

Bucles: Si,(For Loop)

Funcionalidades: Sub Funcionalidad Intercambio Objetos

public static void mostrarInv(int[] inventory)

Función que imprime por pantalla los objetos que se tienen en el inventario

Parámetros: inventory(Tipo Array)

Variables: res(Tipo String) y a(tipo entero)

Condicionales: sí (tipo Switch y If)

Bucles: Si,(For Loop) y

Funcionalidades: Lectura Inventario y Sub-funcionalidad Cambiar Objetos

public static String answers(int juego,int[] ingame)

Función almacena el dígito (en pares) en el array dado según el juego que se juegue(Resumiendo: Guarda la pista encontrada)

Parámetros: juego(Tipo Entero) y ingame(Tipo Array de Enteros)

Variables: res(Tipo String)

Condicionales: sí (Tipo Switch)

Bucles: No

Funcionalidades: Guardar pistas encontradas

public static String settings(int[] ingame)

Función de ajustes principales(Dificultad, "Gráficos"(Esta es cómica, No funcional))

Tabla de contenido

Parámetros: ingame(Tipo Array de enteros)

Variables: settingsInput(Input de tipo entero)(Usado .toLowerCase())

Condicionales: sí (Tipo Switch)

Bucles: Sí(Tipo while)

Funcionalidades: Cambio nivel dificultad

public static String description(int sala)

Función almacena (Para luego devolver) la descripción de la sala correspondiente

Parámetros: sala (Tipo Entero)

Variables: des(Tipo String)

Condicionales: sí (Tipo Switch)

Bucles: No

Funcionalidades:

public static String howto(int lugar)

Función almacena (Para luego devolver) los comandos correspondientes de cada sala

Parámetros: lugar (Tipo Entero)

Variables: instructions(Tipo String)

Condicionales: sí (Tipo Switch)

Bucles: No

Funcionalidades:

public static void muerte()

Función que devuelve que has muerto y sale del juego

Parámetros: No

Tabla de contenido

Variables: No

Condicionales: No

Bucles:No

Funcionalidades:

Nota: Dos posibles finales

public static void exit()

Función que sale del juego

Parámetros: No

Variables: No

Condicionales: No

Bucles: No

Funcionalidades: Salir del juego en cualquier momento

Nota: Extracurricular (System.exit(0)) para terminar el programa

Salas

Todas las salas

public static void sala1(int[] inventory,int[] ingame)(Sara)

public static void sala2(int[] inventory,int[] ingame)(Aída)

public static void sala3(int[] inventory,int[] ingame)(Fernando)

public static void sala4(int[] inventory,int[] ingame)(Aída)

public static void sala5(int[] inventory,int[] ingame)(Fernando)

public static void pasillo(int[] inventory,int[] ingame)(Aída)

Función de Sala (con comandos y opciones a elegir)

Tabla de contenido

Parámetros: respuesta/respuesta/ sala3Input/ sala4Input/ sala5Input(Tipo String y método .toLowerCase() .Se corresponden con las salas respectivamente)

Variables: Sí

Condicionales: Sí(Tipo Switch)

Bucles: Sí (Tipo While)

Funcionalidades: Dependencia/Condiciones durante el juego

`public static void main(String[] args)`

Menú Inicial

Parámetros: args(Array tipo String)

Variables: Sí, menuinput(String)

Condicionales: Sí (Tipo Switch)

Bucles: Sí (Tipo While)

Funcionalidades:

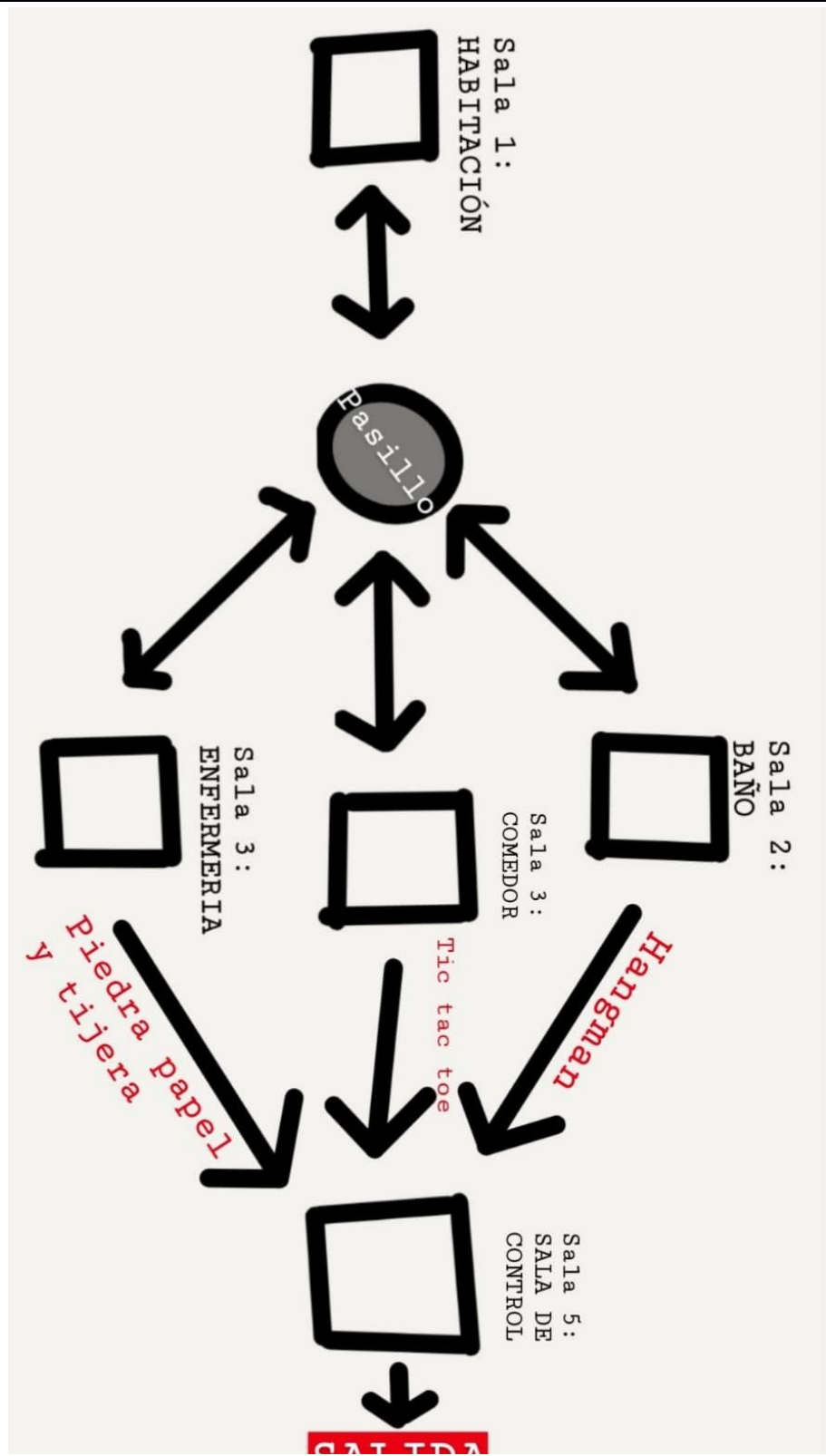
Observación:

Todos los miembros han trabajado en todas las funciones, pese a que los minijuegos fueron desarrollados originalmente por una persona (indicada) y en algunos casos modificados después por el grupo en su conjunto.

Observación 2:

Funciones como la introducción de comandos, las condiciones durante el juego y los dos posibles finales son externas a las funciones mencionadas, ya que se desarrollan durante todo el juego como consecuencias de ciertas acciones o son innatas al juego.

Diagrama del Juego y Diagrama de Decisión



[REDACTED] [REDACTED]

[illegible]

Conclusión

Aída Muñoz:

Ha sido interesante hacer este proyecto, ya que personalmente suponía un reto por su extensión y la novedad de la situación en la que nos encontrábamos mis compañeros y yo.

Mediante la realización de este juego, he descubierto los problemas a los que se enfrenta un programador a la hora de planear un proyecto como este, y la importancia de tener claras las ideas antes de ponerse a teclear. Pese al restringido conocimiento de la materia, estoy contenta con el trabajo que hemos conseguido hacer, aunque estoy segura de que podríamos mejorarlo y ampliarlo con más tiempo.

Sara Beatriz Alonso:

Lo que más me ha gustado es poder trabajar en una idea propia, y creo que nuestra motivación por hacer este trabajo lo mejor que pudiésemos se ha notado en el resultado, a pesar de las dificultades que nos han ido surgiendo y de las limitaciones del tiempo.

Sobre todo estoy muy orgullosa de la dedicación y el cariño de todos los miembros del equipo hacía el proyecto, que en mi opinión han dado como fruto un trabajo muy cuidado y con muchos detalles que te permiten introducirte en la historia a la perfección, para poder identificarte y sentir, en primera persona, a nuestra protagonista.

Fernando Bellido:

Se podría haber mejorado más con algo más de tiempo, el trabajo está bien sin embargo requiere de matices que hay que pulir y añadir más cosas(con más conocimiento de la materia).

Además, de una mejor organización del trabajo puesto que a más líneas más nos costo organizarnos.

Me ha aportado una visión que antes no tenía de cómo se “programaba” en el sentido de las muchas líneas, la organización que todo esto requiere .Admiro lo que me llevo de esta experiencia, una que tenías una idea y acaba en otra muy distinta

Tabla de contenido

Como grupo:

Al principio nos costó coordinarnos, pero pronto nos compaginamos para poder trabajar conjuntamente y sacar adelante el proyecto. Hemos aprendido la importancia de los nombres de las variables usadas, que en muchos casos evitan que el programador se pierda a la hora de escribir, y la importancia de las funciones, que nos han ayudado a estructurar el código de manera más comprensible.

Tabla de contenido

--	--

Tabla de contenido

Webgrafía

<https://www.variablenotfound.com/2008/02/101-citas-clebres-del-mundo-de-la.html>

Referencias

Universidad Politécnica De Madrid

Integrantes Sara Beatriz Alonso Fernández, Aída Muñoz Monjas Fernando Bellido Pazos

Asignatura Programación

Curso Grado Matemáticas e Informática

