

UNIVERSIDAD SIMON BOLIVAR

LAB. ALGORITMO Y ESTRUCTURA I

CI 2691



USB

CLUEDO USB

(Informe de proyecto)

Arleyn Goncalves 10-10290

Francisco Sucre 10-10717

1 DE MARZO DEL 2013

INTRODUCCION

El problema a resolver, es recrear el famoso juego de mesa Cluedo, usando el lenguaje pascal.

El juego se basa en descubrir quien asesino al Dr. Black, con que arma y en que habitación de la mansión Tudor. Solo admite a un jugador y este va a jugar contra las maquinas, solo se permiten entre 3 y 6 jugadores en total. El juego está compuesto por un tablero, y 21 Cartas en la que 6 son personajes, 6 son armas y 9 son habitación que son las mismas habitaciones del tablero. Se elegirán al azar, una carta personaje, una carta arma y una carta habitación, estas se colocan en el sobre del asesinato, el restos de las cartas se barajan y se reparten entre los jugadores. El jugador se va ir desplazando entre las habitaciones y cada vez que llegue a una habitación, el jugador tendra la opción de hacer una sospecha y la opción de acusar, y los otros jugadores intentaran refutar su sospecha mostrando una carta a la persona que hace la sospecha, es el mismo procedimiento con los jugadores maquina solo que la carta a refutar no se va a mostrar.

Cuando el jugador tenga la respuesta a las tres preguntas claves que son: ¿Quién es el culpable?, ¿Cuál es el arma del crimen? y ¿En qué habitación ocurrió el homicidio?, puede hacer una acusación, se verifica la acusación con las del sobre del asesinato, si la acusación es correcta gana y finaliza el juego, y si la acusación es incorrecta pierde y el ganador es la maquina.

El desarrollo de este proyecto nos permitió aplicar los conocimientos adquiridos, y en este informe comentaremos de una manera clara y precisa los procedimientos realizados en cada uno de los subprogramas del programa principal y también explicaremos que realiza cada subprograma.

Además agregaremos una lista de variables y tipos, con una breve explicación de cuál es la función de cada una de ellas y de esta manera hacer mas claro la lectura del proyecto.

DESCRIPCIÓN DE LOS MÓDULOS DE LA CARTA ESTRUCTURADA

PROYECTO USB CLUEDO

En el siguiente documento se presenta una pequeña descripción de la carta estructurada para el proyecto 'USB CLUEDO'

A continuación la descripción de los módulos:

- Cluedo USB -> CluedoUSB

Programa Principal (Ciclo Principal).

- Logo -> Logo

Procedimiento que imprime el logo del juego.

- Introducción -> Introducción

Procedimiento que da la bienvenida al usuario que imprime las instrucciones del juego si el usuario así lo desea.

- Cargar Partida -> Cargar_Partida

Procedimiento que carga un estado de partida.

- Pre Juego -> Prejuego

Procedimiento que permite al usuario definir el número de jugadores, luego elige el personaje de cada jugador y elige las tres cartas culpables, la carta del arma, de personaje y de lugar.

- Inicio -> Inicio

Procedimiento donde se barajan las cartas restantes y se reparten entre los jugadores.

- Tablero -> Tablero

Imprime el tablero antes de que el jugador realice un movimiento.

- Movimiento en el Tablero -> Movimiento

Procedimiento que utiliza los dados para poder calcular la nueva posición del jugador.

- Dado -> Dadoroll

Procedimiento donde se calcula el valor del dado.

- Sospecha (Usuario) -> Sospecha

Procedimiento en el que se formula la pregunta o sospecha del usuario y luego se hace la comparación de la sospecha del usuario con las cartas que tienen los demás jugadores.

- Pregunta (Usuario) -> Pregunta

Procedimiento donde el usuario define su sospecha eligiendo un personaje, un arma y un lugar.

- Todas las Cartas -> Imprimir_Todas_Las_Cartas

Procedimiento donde le imprime al jugador todo el mazo de cartas y así facilitar su decisión para hacer una sospecha

- Mazo -> Imprimir_Mazo

Procedimiento donde se imprime las cartas del jugador.

- Cartas Refutadas -> Refutadas_Mostrar

Procedimiento que imprime las cartas que ya le han sido refutadas al jugador en el juego

- Hipótesis (Usuario) -> Hipótesis

Procedimiento donde el usuario define las cartas de la hipótesis.

- Acusar (Usuario) -> Acusar

El usuario define una hipótesis (formada por un personaje, un arma y un lugar) y compara la acusación con los culpables y verifica si es correcta la acusación.

- Salida Victoria (Usuario) -> SalidaVictoria

Procedimiento del mensaje de salida cuando la acusación es correcta e implica la victoria del usuario y el fin del juego

- Salida Derrota (Usuario) -> SalidaDerrota

Procedimiento del mensaje de salida cuando la acusación es incorrecta e implica la derrota del usuario y el fin del juego.

- Sospecha (Máquina) -> Sospecha_bot

Procedimiento donde la máquina aleatoriamente formula la pregunta o sospecha que no le haya sido refutada antes y luego se hace la comparación de la sospecha con las cartas que tienen los demás jugadores.

- Pregunta (Máquina) -> Pregunta_bot

Procedimiento donde la máquina aleatoriamente define su sospecha eligiendo un personaje, un arma y un lugar.

- Acusar (Máquina) -> Acusar_bot

La máquina formula una hipótesis (formada por un personaje, un arma y un lugar) y compara la acusación con los culpables y verifica si es correcta la acusación.

- Salida Victoria (Máquina) -> SalidaVictoria_bot

Procedimiento del mensaje de salida cuando la acusación es correcta e implica la victoria de la maquina.

LISTA DE VARIABLES Y CONSTANTES

- TIPOS

- **CartGlobal** = (Senora_Blanco, Senora_Celeste, Senorita_Escarlata, Profesor_Ciruela, Coronel_Mostaza, Senor_Verde, Candelabro, Cuchillo, Cuerda, Llave_Inglesa, Revolver, Tubo, Biblioteca ,Cocina, Comedor, Estudio, Vestibulo, Salon, Invernadero, Sala_de_Baile, Sala_de_Billar, Pasillo, Vacio);

Es un tipo enumerado que contiene todas las cartas del mazo, además agrega pasillo y vacio, que a pesar de que no son cartas, son vitales para el desarrollo del programa.

- **Personaje** = Senora_Blanco..Senor_Verde;

Es un tipo subbrando de CartaGlobal, se usa para definir las cartas de personajes.

- **Arma** = Candelabro..Tubo;

Es un tipo subbrando de CartaGlobal, se usa para definir las cartas de armas

- **Lugar** = Biblioteca..Pasillo;

Es un tipo subbrando de CartaGlobal, se usa para definir las cartas de lugar.

- **Direccion** = (Arriba,Abajo,Derecha,Izquierda,Parar)

Es un tipo enumerado que contiene los movimientos que puede ingresar el usuario.

- **Cartas_Jugar** = 1..6

Es el rango de de cartas que puede tener cada personaje.

- **Dado** = 1..6;

Es el rango de valores que puede tener un dado.

- **Posicion** = 1..5;

Es el rango de valores que puede tener la posición de un jugador, tanto horizontal como vertical, teniendo en cuenta que es una matriz 5x5.

- **IdemP** = 1..6;

Es el rango de los números de jugadores.

- **ArrayIdem** = Array[IdemP] of Personaje

Es un array que contiene la identificación de los personajes, asociando un numero

.

- **Afirmacion** = (Si,No)

Es un tipo enumerado que tiene opciones de afirmación.

- **ArrayMov** = Array[Personaje] of Array [Posicion] of Array[Posicion] of Lugar

Es un array que define la habitacion donde se encuentra un personaje, con sus coordenadas verticales y horizontales.

- **ArrayMano** = Array[Personaje] of array[1..21] of CartGlobal

Es un array que contiene las cartas de un personaje

- **ArrayPosicion** = Array[Personaje] of Posicion;

Es un array que contiene la posicion de cada personaje.

- **ArrayCartasUsadas** = Array[1..21] of CartGlobal;

Es un array que contiene una carta del conjunto de CartGlobal

- **Array_Nrefutadas** = Array[Personaje] of Integer;

Es un array que guarda el numero de cartas que le han sido refutadas a cada personaje

- **Conjunto** = set of 1..21;

Es el rango del conjunto de cartas

- **Carta** = record

Arma : Arma;

Lugar : Lugar;

Personaje: Personaje;

Es un tipo record para relacionar una carta con personaje, arma o personaje.

- **ArrayLugarGuardado** = Array[Personaje] of CartGlobal;

Usado para memorizar el lugar de cada personaje al cargar una partida

- **VARIABLES DEL PROGRAMA PRINCIPAL**

- **Culpable**: es tipo carta, es el conjunto de las tres cartas culpables.
- **Mov**: es de tipo ArrayMov, es posición del jugador en el tablero.
- **Mano**: es de tipo ArrayMano, contiene el conjunto de cartas de por jugador.
- **Horizontal_mem**: es de tipo ArrayPosicion, contiene la posición horizontal de cada jugador.
- **Vertical_mem**: es de tipo ArrayPosicion, contiene la posición vertical de cada jugador.
- **ID**: es de tipo IdemP, y contiene el numero que identifica al jugador.

- **Cartas_Usadas:** es de tipo ArrayCartasUsadas, contiene las cartas culpables.
- **Cartas_Usadas2:** es de tipo ArrayCartasUsadas, contiene el mazo de cartas, barajeadas
- **Njugadores:** es de tipo integer, y es el número de jugadores en la partida,
- **Idem:** es de tipo ArrayIdem, contiene el identificador del jugador.
- **Turno:** es de tipo integer, contiene el turno de la partida.
- **Hor:** es de tipo Posicion, contiene la posición horizontal del jugador.
- **Ver:** es de tipo Posicion, contiene la posición vertical del jugador.
- **Pregunta:** es de tipo Carta, es la variable que almacena las respuestas de las sospechas
- **Escena:** es de tipo Lugar, almacena la posición de los jugadores.
- **Particion_Cartas:** es de tipo Carta_jugar, es la cantidad de cartas que va ha tener cada jugador
- **Opcion_sospecha:** es de tipo Afirmacion, es la opción que se le da al jugador de si quiere sospechar o no.
- **Opcion_Acusa:** es de tipo Afirmacion, es la opción que se le da al jugador de si quiere acusar o no
- **Pregunta_Culpable:** es de tipo Carta, es la respuesta de la maquina respecto a su sospecha.
- **Sospechoso:** es de tipo Carta, Variable que almacena al sospechoso elegido por el jugador.

- **Iteracion** : es tipo Integer, Variable de iteracion
- **Codigo**: es de tipo Word, identifica si la información que ingresa el jugador tiene un error.
- **mensajeError**: es de tipo string, es el mensaje de error según el error.
- **Carta_Refutada**: es de tipo integer, es un contador.
- **Refutadas**: es de tipo ArrayMano, contiene las cartas refutadas del jugador.
- **Error_bot_mano**: es del tipo es booleano,
- **Sospecha_Maquina_Arma**: es de tipo CartGlobal, guarda la sospecha maquina del personaje hecha por el jugado-maquina.
- **Sospecha_Maquina_Persona**: es de tipo CartGlobal, guarda la sospecha maquina del arma hecha por el jugado-maquina.
- **Sospecha_Maquina_Lugar**: es de tipo CartGlobal, guarda la sospecha maquina del lugar hecha por el jugado-maquina.
- **Mem_Refutadas**: es de tipo Array_Nrefutadas, guarda las cartas refutadas.
- **Estado_Jugador** : es tipo ArrayEstado, indica si el jugador anda activo o eliminado de la partida
- **Opcion_Cargar** :Es tipo Afirmacion, da la opcion de cargar una partida previamente guardada
- **Lugar_guardado** : Es tipo CartGlobal, guarda el lugar de cada jugador en la partida cargada para asi poder asignar las coordenadas

DECISIONES QUE SE IMPLEMENTARON EN EL PROGRAMA

La implementación del programa se hizo mediante la aplicación de subprogramas, los cuales están mencionados en la descripción de los módulos de la carta estructurada. Están los subprograma de Logo y Tablero que básicamente imprimen el tablero y el logo de Cluedo.

El primer subprograma es el de Introducción, le permite al jugador si iniciar una nueva partida o continuar con una guardada, además le da la opción de imprimir las instrucciones, si el jugador lo desea.

El segundo subprograma es el de Pre juego, el jugador decide el número de jugadores, delimitado entre 3 y 6; al jugador se le permite elegir que personaje quiere ser. A continuación se implementaron una serie de repeat independientes, cada uno va a elegir un personaje por medio de un random y un case, este verifica que el personaje elegido por el random sea diferente al que eligió al jugador y a los randoms anteriores, de esta manera se eligen los personajes que son maquinas. Luego se eligen las cartas culpables, se realiza mediante 3 randoms y dentro de cada random un case, uno para personaje, otro para el arma y otro para el lugar.

El tercer subprograma llamado Inicio, se encarga de barajear las cartas y repartirlas entre el número de los jugadores, para el barajeo de cartas se hizo un random llamado barajeo que puede elegir entre 21 cartas y estas se encuentran dentro de un case, lo siguiente es verificar que la carta elegida no sea ninguna de las culpables esto se hace mediante if, si pasa la condición la carta se guarda en un array vacio denominado números_aleatorio, esto se hace consecutivamente hasta que se elijan 18 cartas y de esta manera cumplir el while que engloba el procedimiento ya explicado.

Luego se hace la repartición de cartas, para lo cual creamos 2 casos principales. El primer caso es en el que sean 3 o 6 jugadores, entonces las cartas se reparten de una manera equitativa sin problema, y el otro caso es el 4 o 5 jugadores, en dicho caso primero se entregan de manera equitativa todas las que se pueden, y con el resto de las

cartas se hace otra ronda de repartición, hasta donde se puede (si sobran 2, se le da 1 al primer jugador y otra al segundo).

El cuarto subprograma se denomina Movimiento, este tiene un subprograma interno llamado Dadoroll, que consta de un random para saber la cantidad de pasos que puede dar el jugador en ese turno, este subprograma se divide en movimientos del jugador y movimientos de la maquina. En movimientos del jugador, el jugador tiene la opción de elegir en qué dirección quiere ir y para guiarse se le imprimirá el tablero, la coordenada en donde se encuentra, el número del dado y la cantidad de pasos que ha dado. Cada vez que el jugador escribe en qué dirección quiere ir, hay una serie de if que verifica que no se salga del tablero, que no quede en un pasillo y actualiza la posición en el tablero. Todo este procedimiento esta englobado por un While que confirma que el número de pasos sea diferente a la cantidad de pasos dados por el jugador.

En movimiento de la maquina, es muy similar al del movimiento del jugador solo que luego de tener el valor del dado, tomara desiciones al azar (Usando una variable a la que se le asigna un random y por medio de "if"s se traducen en movimientos) hasta llegar a una habitacion. Cabe destacar que si alguno de los movimientos tanto del jugador como el de la maquina finaliza en un pasillo, el numero de pasos se reinicia y el personaje se devuelve a la posición inicial.

El Quinto subprograma es denominado Sospecha, este contiene tres subprogramas internos llamados, Pregunta, Imprimir_Mazo, Imprimir_Todas_las_Cartas, Refutadas_Mostrar. El subprograma interno Pregunta, se encarga de preguntar al usuario de quien sospecha, cual es el arma culpable y la habitación es la misma en donde se encuentra. El subprograma interno Imprimi_Mazo, solo imprime las cartas que le fueron repartidas al jugador, el subprograma interno Imprimir_Todas_las_Cartas muestra las cartas del mazo y permite que el usuario conozca todas las cartas y el procedimiento Refutadas_Mostrar muestra las cartas que por la sospecha del usuario hayan sido refutadas por los otros jugadores. En el subprograma general Sospecha, se encarga de verificar por medio de dos for, uno para los jugadores y uno interno para las cartas, que ninguno de los otros jugadores tienen las cartas de la sospecha del usuario, y le muestra al jugador la primera que encuentre, además esta carta que se muestra será mostrada en el subprograma Refutadas_Mostrar.

El Sexto Subprograma es Acusar, este contiene tres subprogramas internos Hipótesis, SalidaVictoria y SalidaDerrota. En el subprograma Hipotesis, el usuario debe escribir a que personaje quiere culpar, el arma con que se cumplió el crimen y la habitación donde ocurrió; en SalidaVictoria se encuentra el mensaje de salida si la acusación es correcta y en SalidaDerrota se encuentra el mensaje de salida si la acusación es incorrecta. En el procedimiento general Acusar, se compara la acusación escrita en Hipotesis con las cartas que se encuentran en el sobre del asesinato si es correcto se imprime el mensaje de SalidaVictoria y si es incorrecto se imprime el mensaje de SalidaDerrota.

El Octavo procedimiento se llama Sospecha_Bot, este tiene un procedimiento interno llamado Pregunta_Bot , este subprograma tiene dos repeat independientes, cada repeat tiene un random interno, en uno se elige un sospechoso y en el otro se elige el arma del crimen, dentro de cada repeat se verifica que la carta elegida no se encuentre en su mazo y no se encuentre dentro de las cartas refutadas. El procedimiento Sospecha_bot es similar al de Sospecha del usuario, tiene dos for y se verifica si alguno de los jugadores tiene alguna de las cartas para refutar su sospecha, solo se muestra una carta y esta se almacena en las cartas refutadas del jugador maquina.

El noveno y último procedimiento Acusa_Bot, que tiene el subprograma SalidaVictoria_Bot que imprime el mensaje de salida cuando la acusación es correcta. El subprograma Acusa_Bot imprime las acusaciones del jugador maquina; para poder ingresar a este subprograma, en el programa principal CluedoUsb se verifica que la sospecha hecha por el jugador maquina no sea refutada por ninguno de los otros jugadores y ninguna de las cartas de la sospecha se encuentre en su mano de cartas, asegurando su acusación.