

Trabajo Práctico en Etapas

Juego Memotest – Parte 1

Objetivo

Construir un programa en etapas, que permita jugar al Memotest, teniendo en cuenta las reglas que se detallan en cada una de las etapas.

Instrucciones Adaptadas del Juego a un único participante

- El Memotest se compone de fichas repetidas de a pares, que mezcladas se colocan sobre la mesa con la figura hacia abajo formando un cuadro.
- El participante comienza dando vuelta primero una ficha y luego otra tratando que sean iguales, si lo consigue toma las dos fichas para sí.
- Si las figuras son distintas las vuelve a dar vuelta en la misma posición que se encontraban y vuelve a elegir un nuevo par.
- El juego finaliza cuando no quedan más fichas por dar vuelta.

Es recomendable que juegues para que te familiarices con la operatoria, a continuación, encontrarás un enlace a un sitio que te permitirá jugar online.

<https://cardsimages.info-tuparada.com/1143/20172-2-juego-memotest.html>

Etapas 1- Versión reducida

En esta etapa el objetivo es programar una pequeña parte del juego.

Vamos a simplificar y adaptar la programación de nuestro juego a los siguientes requerimientos:

- Las fichas por adivinar sólo serán 4, o sea 2 pares de fichas iguales.
Por ejemplo: "D", "D", "s", "s".
- El contenido de cada ficha será una letra, por ejemplo, las letras: D (mayúscula) y s (minúscula). Cada una de las cuales estará asociada a una posición en nuestro tablero de 4 fichas.

- Se le mostrará al jugador las posiciones disponibles, y se le solicitará el ingreso de una de ellas. Algo similar al siguiente ejemplo:

Fichas y Posiciones: [1] [2] [3] [4]

1er. Posición: _

En esta etapa, no debemos preocuparnos por validar el ingreso de los valores.

- A continuación, debemos mostrar que letra hay en la posición ingresada, y solicitar el ingreso de la siguiente posición en la que supone estará la letra gemela.
Por ejemplo si el jugador había ingresado como 1er. Posición, 2; entonces

veríamos algo parecido a lo siguiente:

Fichas y Posiciones: [1] D [3] [4]

2da. Posición: _

- Si la segunda posición elegida contiene la letra gemela de la primer posición, entonces se mostrará al usuario la dos posiciones con sus respectivas letras, y se volverá a solicitar una primer posición:

Fichas y Posiciones: [1] D [3] D

1er. Posición: _

- En cambio, si la posición elegida como segunda posición, no contiene la letra gemela de la primera posición, se vuelve a mostrar el tablero con todas las posiciones disponibles:

Fichas y Posiciones: [1] [2] [3] [4]

1er. Posición: _

- Esta etapa del juego termina, cuando el usuario adivina todos los pares de posiciones con letras gemelas.

Etapa 2 - Validación de ingresos

En esta etapa, sumaremos a los requerimientos de la etapa anterior, el de validar el ingreso de las posiciones.

Si el participante ingresa:

- un valor que no corresponda a una posición
- o el mismo no se encuentre disponible,
- o no se trata de un valor numérico,

se le debe mostrar un mensaje acorde y solicitar el ingreso de un nuevo valor.

Etapa 3 - Aumentando y generalizando la cantidad de fichas

En esta etapa debemos aumentar la cantidad de fichas intervinientes.

Sumaremos a los requerimientos de la etapa anterior los siguientes:

- La cantidad de fichas intervinientes será ahora de 8 fichas, o sea 4 pares de fichas iguales.
- Debemos mostrar 4 fichas por línea.
- Además, debemos informar al terminar el juego, cuantos intentos tuvo que realizar el participante para terminar la partida.

Etapas 4 - Aleatoriedad y tiempos

Hasta aquí ya hemos logrado que nuestro juego se parezca bastante al real, por ello le daremos unos toques finales, sumando los siguientes requerimientos:

- Es hora de agregar **aleatoriedad a las ubicaciones de las fichas**. En cada ejecución del juego, debemos lograr que nuestras fichas sean asignadas a distintas posiciones.
- También, debemos **informar el tiempo que ha tomado la partida**.
- Por último, es hora de ampliar la cantidad de fichas a 16, mostrando 4 por línea.

Etapas 5 - Extensión a Dos participantes

En el juego de mesa, los participantes pueden ser 2 ó más. Lo que cambia respecto de lo que hemos hecho, es lo siguiente:

- Comienza un participante dando vuelta primero una ficha y luego otra tratando que sean iguales, si lo consigue toma las dos fichas para sí.
- Si las figuras son distintas las vuelve a dar vuelta en la misma posición que se encontraban y entonces dar lugar al otro participante a que haga lo mismo; y así sucesivamente.
- El juego finaliza cuando no quedan más fichas y se declara ganador al jugador con mayor cantidad de fichas iguales. En caso de empate, ganará el jugador que tuvo menor cantidad de intentos.

Entonces, en esta etapa debemos dar la posibilidad a que el juego sea entre dos participantes. Para ello, debemos:

- Solicitar los nombres de los dos jugadores al inicio de la partida.
La solicitud de los nombres se debe hacer mediante una interfaz gráfica.
- Elegir de forma aleatoria, el jugador que jugará primero; y antes del comienzo del juego informar el orden. Cada vez que un jugador recibe su turno, se debe indicar claramente "Turno de: (Nombre del Jugador)".
- Adaptar nuestra aplicación para que la lógica siga con esta nueva dinámica de juego. O sea, cuando un jugador no acierta la posición de la segunda ficha, le tocará el turno de elección de la posición de la primera ficha, al siguiente jugador.

Condiciones de Entrega

Las siguientes condiciones deben ser respetadas para que la entrega sea considerada válida:

1. El programa debe ser desarrollado en Python, aplicando correctamente los conceptos dados en clase y respetando las buenas prácticas descriptas en el PEP de la cátedra.
Cada función que forma parte del código debe tener debajo de su firma, una descripción corta de cuál es su objetivo y quien es el autor ó responsable de dicha función.
2. El código correspondiente a la Parte 1, debe ser subido al campus. El nombre a dar al archivo será `TP1_NombreGrupo.py`. Deberán reemplazar *NombreGrupo*, por el nombre dado a su grupo. Si la entrega está compuesta por más de un archivo .py, generar un .zip con todos los archivos .py, y nombrarlo de igual modo, pero con extensión zip.
3. Deberán grabar 2 videos y subirlos a un canal de Youtube, ó a Google Drive. El primer video, cada integrante del equipo, deberá contar mostrando el código, qué parte estuvo bajo su responsabilidad y los puntos de solución dados, que considere más relevantes. El video total no debe superar los 10 minutos. Comenzar cada uno de los relatos, diciendo el nombre y apellido del quien relata. La calidad del video debe ser buena, y debe haber homogeneidad en las presentaciones.
4. Deberán grabar un segundo video, en el que se muestre al menos una jugada completa, y que contemple distintos casos que muestran que la aplicación responde según lo esperado. Deberán ir relatando los eventos de la jugada. En este caso el video puede estar realizado por 1 único integrante.