

Universidad de San Carlos de Guatemala

Centro Universitario de Occidente

División de Ciencias de la Ingeniería

Área Profesional

Introducción a la Programación y Computación 1

Sección "A"

Ing. José Moisés Granados Guevara



“MANUAL TECNICO PRACTICA 1”

Melvin Eduardo Ordoñez Sapón RA | 202230552

Quetzaltenango, febrero del 2024

“Id y enseñad a todos”

ACERCA DEL PROGRAMA:

IDE utilizado: Apache NetBeans IDE 20

JDK utilizado: Versión 21.0.2.0

Programado en Sistema Operativo: Windows 11 Pro

Terminal Utilizada: Terminal de Microsoft Store

Lenguaje de Programación Utilizado: Java

Compatibilidad: Windows, Linux, macOS

ALGORITMOS IMPORTANTES DE LOS JUEGOS

“VARIABLES GLOBALES”

```
const RESET = "\u001B[0m"  
const VERDE = "\u001B[32m"  
const AMARILLO = "\u001B[33m"  
const ROJO = "\u001B[31m"  
const AZUL = "\u001B[34m"  
const GRIS = "\u001B[90m"  
const CELESTE = "\u001B[96m"  
const NARANJA = "\u001B[33m"
```

```
var wordleIniciados = 0  
var wordlePerdidos = 0  
var wordleGanados = 0  
var intentosadivinarPalabra = 0
```

```
var basketballIniciados
```

```
var veintecuatrochoIniciados  
var veintecuatrochoGanado  
var veintecuatrochoSalir
```

```
var juegoActual = ""
```

METODO MAIN

si (longitud(args) > 0) entonces

 // Convierte el parámetro enviado a minúsculas.

 cadena parametro = a_minuscula(args[0])

 seleccionar (parametro)

 caso "wordle":

 imprimir("EL JUEGO SE HA INICIADO CON EL PARAMETRO: " +
parametro)

 wordleIniciados = wordleIniciados + 1

 llamar a Wordle()

 romper

 caso "basketball":

 imprimir("EL JUEGO SE HA INICIADO CON EL PARAMETRO: " +
parametro)

 basketballIniciados = basketballIniciados + 1

 llamar a Basketball()

 romper

 caso "2048":

 imprimir("EL JUEGO SE HA INICIADO CON EL PARAMETRO: " +
parametro)

 veintecuatrochoIniciados = veintecuatrochoIniciados + 1

 llamar a iniciarJuego2048()

 romper

 defecto:

 fin seleccionar

fin si

llamar a menuPrincipal()

MENU PRINCIPAL

procedimiento menuPrincipal()

// MENU PRINCIPAL

iniciar scanner = nuevo Scanner(System.in)

entero opcion = 0

hacer

imprimir("MENU PRINCIPAL")

imprimir("1.Wordle")

imprimir("2.Basketball")

imprimir("3.2048")

imprimir("4.Reportes")

imprimir("5.Salir")

opcion = convertir_a_entero(scanner.leerLinea())

seleccionar (opcion)

caso 1:

wordleIniciados = wordleIniciados + 1

llamar a Wordle()

limpiarPantalla()

romper

caso 2:

basketballIniciados = basketballIniciados + 1

llamar a Basketball()

limpiarPantalla()

romper

caso 3:

veintecuatroochoIniciados = veintecuatroochoIniciados + 1

llamar a iniciarJuego2048()

limpiarPantalla()

romper

caso 4:

limpiarPantalla()

llamar a reportes()

romper

caso 5:

llamar a finalizarPrograma()

romper

defecto:

limpiarPantalla()

imprimir("Elija una opcion Valida")

fin seleccionar

mientras (opcion != 5)

fin procedimiento

“WORDLE”

función verificadorPalabra(palabraOculto: cadena)

 si longitud(palabraOculto) == 5 entonces

 // LA PALABRA INGRESADA ES DE 5 CARACTERES O MENOS

 devolver verdadero

 sino

 imprimir("INGRESE UNA PALABRA DE 5 CARACTERES")

 devolver falso

 fin si

fin función

función palabraAnterior(palabraOculto: cadena, letras: arreglo de caracteres, adLetras:
arreglo de caracteres)

 variable resultado = nueva CadenaVacía()

 para i desde 0 hasta 4 hacer

 si letras[i] es igual a adLetras[i] entonces

 imprimir(VERDE + adLetras[i] + RESET)

 sino si palabraOculto contiene convertir_a_cadena(adLetras[i]) entonces

 imprimir(AMARILLO + adLetras[i] + RESET)

 sino

 imprimir(ROJO + adLetras[i] + RESET)

 fin si

 fin para

 devolver resultado

fin función

“BASKETBALL”

función lanzamiento(parametro: entero)

variable aleatorio = generar_aleatorio_entre(0, 100)

si aleatorio < parametro entonces

imprimir("Lanzamiento exitoso")

devolver verdadero

sino

imprimir("Lanzamiento fallido")

devolver falso

fin si

fin función

función tiroLibre() entero

// 90% DE PROBABILIDAD DE ANOTAR LA JUGADA

si lanzamiento(90) entonces

imprimir("Tiro libre anotado")

devolver 2

sino

imprimir("Tiro libre fallado :(")

devolver 0

fin si

fin función


```

función accionValida(accion: entero)
    si accion es igual a 1 o accion es igual a 2 entonces
        devolver verdadero
    sino
        imprimir("Ingrese una opción válida")
        devolver falso
    fin si
fin función

```

JUEGO 2048

```

procedimiento bulclePartida2048(tamfilas: entero, tamcolumnas: entero, tablero: arreglo
de arreglos de enteros)
    // Inicializar scanner y dirección
    iniciar scanner = nuevo Scanner(System.in)
    cadena direccion

    hacer
        imprimir("          Arriba (w)")
        imprimir("Izquierda(a)  Abajo (s)  Derecha(d) "+ROJO+"Abandonar
Partida(p)"+RESET)
        direccion = scanner.leerLinea()
        imprimir("\n\n")

    seleccionar (direccion)
        caso "w":
            moverTodoHaciaArriba(tamfilas, tamcolumnas, tablero)
            mostrarTablero(tamfilas, tamcolumnas, tablero)
            romper

```

caso "s":

moverTodoHaciaAbajo(tamfilas, tamcolumnas, tablero)

mostrarTablero(tamfilas, tamcolumnas, tablero)

romper

caso "a":

moverTodoHaciaLaIzquierda(tamfilas, tamcolumnas, tablero)

mostrarTablero(tamfilas, tamcolumnas, tablero)

romper

caso "d":

moverTodoHaciaLaDerecha(tamfilas, tamcolumnas, tablero)

mostrarTablero(tamfilas, tamcolumnas, tablero)

romper

caso "p":

imprimir("Ha decidido salir de la partida")

veintecuatrochoSalir = veintecuatrochoSalir + 1

romper

defecto:

imprimir("Elija una opción válida")

fin seleccionar

mientras no (direccion es igual a "p") y no finPartida(tamfilas, tamcolumnas, tablero)

si finPartida(tamfilas, tamcolumnas, tablero) entonces

imprimir("")

si ganador(tamfilas, tamcolumnas, tablero) entonces

veintecuatrochoGanado = veintecuatrochoGanado + 1

imprimir(AZUL+"***Has ganado***"+RESET)

```

    sino
        imprimir(ROJO+"***Has Perdido***"+RESET)
    fin si
fin si
fin procedimiento

```

procedimiento crearTablero(tamfilas: entero, tamcolumnas: entero, tablero: arreglo de arreglos de enteros)

```

    para filas desde 0 hasta tamfilas - 1 hacer
        para columnas desde 0 hasta tamcolumnas - 1 hacer
            tablero[filas][columnas] = 0
        fin para
    fin para
fin procedimiento

```

procedimiento mostrarTablero(tamfilas: entero, tamcolumnas: entero, tablero: arreglo de arreglos de enteros)

```

    para filas desde 0 hasta tamfilas - 1 hacer
        para columnas desde 0 hasta tamcolumnas - 1 hacer
            si tablero[filas][columnas] > 1000 entonces
                imprimir("|" + ROJO + " " + tablero[filas][columnas] + " " + RESET)
            sino si tablero[filas][columnas] > 100 entonces
                imprimir("|" + AMARILLO + " " + tablero[filas][columnas] + " " + RESET)
            sino si tablero[filas][columnas] > 10 entonces
                imprimir("|" + VERDE + " " + tablero[filas][columnas] + " " + RESET)
            sino si tablero[filas][columnas] > 0 entonces
                imprimir("|" + AMARILLO + " " + tablero[filas][columnas] + " " + RESET)
            sino
                imprimir("|" + CELESTE + " " + tablero[filas][columnas] + " " + RESET)
        fin para
    fin para

```

```

        fin si
    fin para
    imprimir("|")
    imprimir("\n")
fin para
fin procedimiento

```

función ganador(tamfilas: entero, tamcolumnas: entero, tablero: arreglo de arreglos de enteros)

```

    para fila desde 0 hasta tamfilas - 1 hacer
        para columna desde 0 hasta tamcolumnas - 1 hacer
            si tablero[fila][columna] == 2048 entonces
                devolver verdadero
            fin si
        fin para
    fin para
    devolver falso
fin función

```

procedimiento numeroDosOrCuatroAleatorio(tamfilas: entero, tamcolumnas: entero, tablero: arreglo de arreglos de enteros)

```

    // Crear un objeto Random
    variable random = nuevo Random()
    entero rndFila
    entero rndColumna

    hacer
        rndFila = random.nextInt(tamfilas)
    mientras (vacíaEnFila(rndFila, tamcolumnas, tablero) == 0)

```

```

hacer
    rndColumna = random.nextInt(tamcolumnas)
    mientras (tablero[rndFila][rndColumna] != 0)

// Generar un número aleatorio para determinar si es 2 o 4
entero dosOcuatro = random.nextInt(2)

seleccionar (dosOcuatro)
    caso 0:
        imprimir(AZUL + "-----" +
RESET)
        imprimir("Número 2 generado en la fila: " + rndFila + " Columna: " +
rndColumna)
        imprimir(AZUL + "-----" +
RESET)
        tablero[rndFila][rndColumna] = 2
        romper
    caso 1:
        imprimir(AZUL + "-----" +
RESET)
        imprimir("Número 4 generado en la fila: " + rndFila + " Columna: " +
rndColumna)
        imprimir(AZUL + "-----" +
RESET)
        tablero[rndFila][rndColumna] = 4
        romper
    defecto:
        lanzar_error("Error inesperado")
fin seleccionar
fin procedimiento

```

procedimiento moverTodoHaciaArriba(tamfilas: entero, tamcolumnas: entero, tablero:
arreglo de arreglos de enteros)

```
para columna desde 0 hasta tamcolumnas - 1 hacer
    colocarArriba(tamfilas, tamcolumnas, tablero, columna)
    sumarHaciaArriba(tamfilas, tamcolumnas, tablero, columna)
    colocarArriba(tamfilas, tamcolumnas, tablero, columna)
fin para
```

// Luego de realizar el movimiento, se genera un número aleatoriamente en una casilla
vacía "0"

```
// Verificar si hay espacio para generar un dos automáticamente
si no finPartida(tamfilas, tamcolumnas, tablero) entonces
    numeroDosOrCuatroAleatorio(tamfilas, tamcolumnas, tablero)
fin si
fin procedimiento
```

procedimiento moverTodoHaciaAbajo(tamfilas: entero, tamcolumnas: entero, tablero:
arreglo de arreglos de enteros)

```
para colocar desde 0 hasta tamcolumnas - 1 hacer
    colocarAbajo(tamfilas, tamcolumnas, tablero, colocar)
    sumarHaciaAbajo(tamfilas, tamcolumnas, tablero, colocar)
    colocarAbajo(tamfilas, tamcolumnas, tablero, colocar)
fin para
```

```
si no finPartida(tamfilas, tamcolumnas, tablero) entonces
    numeroDosOrCuatroAleatorio(tamfilas, tamcolumnas, tablero)
fin si
fin procedimiento
```

procedimiento moverTodoHaciaLaDerecha(tamfilas: entero, tamcolumnas: entero,
tablero: arreglo de arreglos de enteros)

para colocar desde 0 hasta tamfilas - 1 hacer

colocarDerecha(tamfilas, tamcolumnas, tablero, colocar)

sumarDerecha(tamfilas, tamcolumnas, tablero, colocar)

colocarDerecha(tamfilas, tamcolumnas, tablero, colocar)

fin para

si no finPartida(tamfilas, tamcolumnas, tablero) entonces

numeroDosOrCuatroAleatorio(tamfilas, tamcolumnas, tablero)

fin si

fin procedimiento

procedimiento moverTodoHaciaLaIzquierda(tamfilas: entero, tamcolumnas: entero,
tablero: arreglo de arreglos de enteros)

// 'colocar' indicará las filas sobre las que se operará

para colocar desde 0 hasta tamfilas - 1 hacer

moverIzquierda(tamfilas, tamcolumnas, tablero, colocar)

sumarIzquierda(tamfilas, tamcolumnas, tablero, colocar)

moverIzquierda(tamfilas, tamcolumnas, tablero, colocar) // Para rellenar posibles

espacios vacíos

fin para

si finPartida(tamfilas, tamcolumnas, tablero) entonces

numeroDosOrCuatroAleatorio(tamfilas, tamcolumnas, tablero)

fin si

fin procedimiento