Universidad de Costa Rica
I Semestre, 2019

Ciencias de la Computación e Informática
Introducción a la Computación – CI0110

Prof. Maureen Murillo
Grupo 03

Documentación del proyecto I-2019
Juego “15 en fichas”

Fabián Orozco Chaves - B95690
Dominic Tarassenko - B97790

Objetivos.

1. Introducir al estudiante en el mundo de la programación.
2. Poner en práctica los conocimientos teóricos adquiridos en el curso sobre estructuras de control y elaboración de algoritmos utilizando un ambiente de programación con un nivel de abstracción alto para la creación de aplicaciones, específicamente en Python.

Descripción de la aplicación

La aplicación permite al usuario jugar contra la computadora el juego “15 en fichas”. En este juego se dispone de una banda numérica del 1 al 9 y de 3 fichas rotuladas con una “u” (para el usuario) y 3 fichas rotuladas con una “c” (para la computadora). Cada jugador por turno pone una de sus fichas sobre una casilla libre o mueve una si ya colocó las 3. En cada casilla sólo puede haber una ficha a la vez. El primer jugador que sume 15 con sus 3 fichas colocadas en la banda es el ganador. No está permitido “saltarse el turno”, siempre hay que poner una ficha si está disponible o moverla si no.

Si a la computadora se le presenta una oportunidad para ganar en un movimiento, siempre lo hará. De no ser posible, intentará bloquear los espacios que le permiten al usuario ganar en un turno. Más allá de eso, la computadora no sabe cuáles pueden ser las consecuencias de sus movimientos, y mueve casi aleatoriamente si no se encuentra en una de las situaciones mencionadas anteriormente.

Diseño de la aplicación

```
def colocarFichasIniciales(turnoActual,primerTurno,tablero){
   si (primerTurno == 1 y turnoActual%2 != 0) o (primerTurno == 2 y turnoActual%2
== 0){ #Pone el usuario
         valido = 0
         mientras valido == 0{
               mostrarTablero(tablero)
               eleccion = entrada("Escriba el número del espacio en el que quiere
poner su ficha")
               si eleccion == 'q'{
                     devolver 1
               {
               sino{
                     si eleccion == 'i'{
                           mostrarInstrucciones()
                     }
                     sino{
                           si eleccion >=1 y eleccion <= 9{
                                 destinoFicha = eleccion
                                 si tablero[destinoFicha-1] == "{
                                       tablero[destinoFicha-1] = 'u'
                                       valido = 1
                           }
                                 sino{
                                       imprimir("Ese espacio está ocupado,
pruebe de nuevo")
                                 }
                           }
                           sino{
                                 imprimir("Opcion invalida, pruebe de nuevo")
                           }
                     }

               }
         }
   }
   sino{ #Pone la computadora
         si largo(calcularCasosFavorables('c',tablero,turnoActual)) > 0{
               victoriaAsegurada = calcularCasosFavorables('c',tablero,turnoActual)
               tablero[victoriaAsegurada[0]-1] = 'c'
               imprimir("La computadora pone en ", victoriaAsegurada[0])
```

```
        }
        sino{
                si calcularCasosFavorables('u',tablero,turnoActual){
                        evitarDerrota =
calcularCasosFavorables('u',tablero,turnoActual)
                        para i en rango(0,largo(evitarDerrota)){
                                si tablero[evitarDerrota[i]-1] == '':
                                        tablero[evitarDerrota[i]-1] = 'c'
                                        print("La computadora pone en ", evitarDerrota[i])
                                        break
                        }
                }
                sino{
                        fichaPuesta = Falso
                        mientras fichaPuesta == Falso{
                                si turnoActual <= 2{
                                        destinoFicha = enteroAleatorio(7,9)
                                }
                                sino{
                                        destinoFicha = enteroAleatorio(1,4)
                                }
                                si tablero[destinoFicha-1] == '{
                                        tablero[destinoFicha-1] = 'c'
                                        imprimir("La computadora pone en", destinoFicha)
                                        fichaPuesta = Verdadero
                                }
                        }
                }
        }
   }
   devolver 0
}

###############Programa principal#########################################

imprimir("Bienvenido a 15 en fichas.")
mensajeBienvenida = "1. Jugar 2. Ver instrucciones 3. Salir. Elección: "
vecesJugadas = 0
victoriasUsuario = 0
victoriasComputadora = 0
salir = 0
mientras salir != 1{
```

```
    si vecesJugadas > 0{
        imprimir("Victorias")
        imprimir("Usuario: ", victoriasUsuario)
        imprimir("Computadora: ", victoriasComputadora)
    }
    eleccionMenu = entrada(mensajeBienvenida)
    turnoActual = 1
    si eleccionMenu == 1{
        primerTurno = 0
        mientras primerTurno != 1 y primerTurno != 2{
            primerTurno = entrada("Escriba 1 para ir de primero o 2 para ir de
segundo: ")
        }
        tablero = ["","","","","","","","","]
        juegoTerminado = 0
        mientras juegoTerminado == 0{
            imprimir("Turno:", turnoActual)
            si turnoActual <= 6{
                juegoTerminado =
colocarFichasIniciales(turnoActual,primerTurno,tablero)
                turnoActual += 1
            }
            sino{
                juegoTerminado =
moverFichas(turnoActual,primerTurno,tablero)
                turnoActual+=1
            }
            si comprobarVictoria(tablero,turnoActual) == 1{
                mostrarTablero(tablero)
                si comprobarVictoria(tablero,turnoActual) == 1{
                    print("Gana el usuario")
                    victoriasUsuario = victoriasUsuario + 1
                }
                sino{
                    print("Gana la computadora")
                    victoriasComputadora = victoriasComputadora + 1
                }
                vecesJugadas = vecesJugadas + 1
                juegoTerminado = 1
            }
        }
    }
```

```
    sino{
        si eleccionMenu == 2{
            mostrarInstrucciones()
        }
        sino{
            si eleccionMenu == 3{
                salir = 1
            }
            sino{
                print("Escoja una opción válida")
            }
        }
    }
}
```

Dificultades encontradas y limitaciones

Aunque la computadora es capaz de ver los espacios con los que ella o el usuario pueden ganar el próximo turno, no se da cuenta de si sus movimientos pueden crear oportunidades para el usuario. Solo puede analizar el tablero actual y no “pensar” en el futuro.

Usabilidad

1. **Visibilidad del estado del sistema:** El programa muestra la cinta, el turno actual, además de los movimientos del usuario y de la computadora en pantalla cada turno. El usuario siempre sabe el estado del tablero y lo último que ha pasado.
2. **Relación entre el sistema y el mundo real**: Todas las instrucciones, al igual que el estado del tablero, se le dan al usuario en un lenguaje natural con el que está familiarizado.
3. **Control y libertad del usuario**: el usuario no puede deshacer sus jugadas, sin embargo, tiene la posibilidad de abortar cualquier partida en juego y volver al menú escribiendo q.
4. **Consistencia y estándares**: la interfaz no tiene mucho texto, pero lo que se le muestra al usuario se puede interpretar de una única manera. No se utilizan varias palabras para hablar de una misma cosa, y no se utilizan palabras que pueden tener varios significados.

5. **Prevención de errores**: en cada oportunidad que tiene el usuario de ingresar datos al programa, si éste ingresa uno que no es válido, se le hace saber y se le da una segunda oportunidad. Este mensaje de error es inevitable pues en una interfaz de texto el usuario puede ingresar el texto que quiera, pero no interfiere con la ejecución del programa.
6. **Reconocimiento antes que recuerdo**: el programa muestra la cinta con su estado actual, los turnos y las acciones que vayan tomando tanto el usuario como la computadora en la pantalla cada turno. Así el usuario tiene frente a él toda la información que necesita para jugar su turno, sin tener que memorizar nada. También, de hacer falta, el usuario puede pedir que se le muestren las instrucciones las veces que sean necesarias escribiendo el caracter i en lugar de su movimiento.
7. **Flexibilidad y eficiencia de uso**: lo único que se podría considerar un acelerador es la posibilidad de ingresar i para ver las instrucciones en media partida sin tener que regresar al menú principal. Es un juego muy simple que no requiere atajos.
8. **Estética y diseño minimalista**: en todo momento al usuario solo se le presenta información actualmente relevante para él.
9. **Ayudar a los usuarios a reconocer**, **diagnosticar y recuperarse de errores:** cuando el usuario ingresa el valor de una ficha que ya está en uso se le muestra un mensaje claro de que el espacio está ocupado y se le pide que lo vuelva a intentar. Con esto se logra que el usuario sepa qué hizo “mal” y lo pueda arreglar.
10. **Ayuda y documentación**: tanto en el menú principal como durante la partida se pueden desplegar las instrucciones, que sirven de documentación. La sección de instrucciones está separada en las siguientes secciones: descripción, objetivo, reglas y controles.