



PRÀCTICA 1.1 : FONAMENTS DE SISTEMES OPERATIUS

Carlos Martínez i Genís Martínez

GEI
2021-2022

Pràctica avaluable



UNIVERSITAT
ROVIRA I VIRGILI

Índex

Índex.....	2
1.- Decissions de disseny	3
2.- Pla de proves	4
3.- Codi font, conclusions i autoavaluació.....	5

1.- *Decisions de disseny*

Per a realitzar aquesta pràctica el primer que vam fer va ser repasar conceptes sobre “bash” explicats als primers laboratoris i investigar sobre la programació estructurada en funcions. Un cop realitzat això vam decidir dissenyar diverses funcions encarregades de subdividir les tasques que s’haurien de realitzar per al correcte funcionament de la pràctica.

Aquestes són:

- **readFile** - llegeix les dades del fitxer
- **writeFile** - escriu les dades al fitxer
- **introducirRestantes** - indica quins elements falten per introduir
- **printVariables** - llista els valors de les variables
- **comprobarValores** - comprova que els valors de les variables estiguin dins dels rangs correctes
- **comprobarFichero** - comprova que existeix el fitxer i trosseja les variables
- **readFromKeyboard** - segons les variables que el “getops” detecti que falten, les demana per teclat

Un cop encapsulades les tasques en aquestes funcions, a la part inferior de tot el codi fem les crides en l’ordre lògic per a que el programa funcioni correctament. Comprovem l’existència d’un arxiu anterior, comprovem els valors, introduïm els restants, etc...

2.- Pla de proves

Per a comprovar que la nostra solució al problema és funcional, hem seguit la comanda d'exemple proporcionada al guió de la pràctica amb diferents valors. Per exemple:

```
./configuracio -n nomFit -f 25 -c 40 -p 20 -o 22,30 -m 10 -l 21,20,-1.0,0.3 ,4,5,-1.0,0.3
```

Ademés:

- Hem anat canviant els valors de les variables per comprovar que es guardaven correctament.
- Hem creat fitxers amb posicions buides per veure com es comportava el programa.
- Hem creat fitxers amb variables fora de rang.
- Hem introduït per comanda variables fora de rang.
- Hem introduït per comanda un nom de fitxer inexistent.

Prova d'exemple:

```
milax@casa:~/Documents/GitHub/FSOPrac1$ ./Prac1.sh -n paramsProva.txt -f 20 -c 20 -p 19 -o 22,30 -m 10 -l 21,20,-1.0,0.3 ,4,5,-1.0,0.3
BIENVENIDO/A AL PROGRAMA
Option n has been chosen
Option f has been chosen
Option c has been chosen
Option p has been chosen
Option m has been chosen
*****
encara ens falta tractar els següents elements:
,4,5,-1.0,0.3
*****

COMPROBAR FICHERO
25
30
20
22
30
10
21
20
-1.0
0.3
COMPROBAR VALORES

VALORES DE LAS VARIABLES:
fils: 25
cols: 30
portSize: 20
posFilaPal: 22
posColPal: 30
midaPaleta: 10
posFilaPil: 21
posColPil: 20
velFil: -1.0
velCol: 0.3
WRITE FILE
INTRODUCIR VALORES RESTANTES
milax@casa:~/Documents/GitHub/FSOPrac1$
```

3.- Codi font, conclusions i autoavaluació

Codi font:

```
#!/bin/bash
fil=0
cols=0
portSize=0

minFil=10
maxFil=120

minCol=8
maxCol=36

minPort=8

minPort=8
temp=fil
posFilaPal=0
posColPal=0

posFilaPil=0
posColPil=0
velFil=0
velCol=0

Nflag=''
Fflag=''
Cflag=''
Pflag=''
OFlag=''
OneFlag=''
MFlag=''

found='false'
extraPilotes='false'
function readFile
{
    echo "READ FILE"
    #falta arreglar, sólo lee si hay 2 líneas en el archivo
    echo "Entra en 'readFile'"
    while IFS=' ' read -r fil cols portSize
    do
        echo "fils= $fil"
        echo "cols= $cols"
        echo "portSize= $portSize"
```

```

        exit
    done < $file
}

function writeFile
{
    echo "WRITE FILE"
    echo "$fil $cols $portSize" > $file
    echo "$posFilaPal $posColPal $midaPaleta" >> $file
    echo "$posFilaPil $posColPil $velFil $velCol" >>$file
}

function introducirRestantes
{
    echo "INTRODUCIR VALORES RESTANTES"
    if [ -z $fil ] || [ -z $cols ] || [ -z $portSize ] ; then
        read -p "Introduce de nuevo las filas (10,120): " fil
        read -p "Introduce de nuevo las columnas (10,36): " cols
        let maxS=fil-1
        read -p "Introduce de nuevo el tamaño de la portería (8, $maxS): " portSize
    fi

    if [ -z $posFilaPal ] || [ -z $posColPal ] || [ -z $midaPaleta ] ; then
        read -p "Introduce de nuevo posFilaPal (1,118): " posFilaPal
        read -p "Introduce de nuevo posColPal (2,35): " posColPal
        read -p "Introduce de nuevo midaPaleta (3,$maxS): " midaPaleta
    fi

    if [ -z $posFilaPil ] || [ -z $posColPil ] || [ -z $velCol ] || [ -z $velCol ];
then
        read -p "Introduce de nuevo posFilaPil (2,118): " posFilaPil
        read -p "Introduce de nuevo posColPil (2,35): " posColPil
        read -p "Introduce de nuevo velFil (-1.0-1.0): " velFil
        read -p "Introduce de nuevo velCol (-1.0-1.0): " velCol
    fi
}

function printVariables
{
    echo ""
    echo "VALORES DE LAS VARIABLES:"
    echo "fils: $fil"
    echo "cols: $cols"
    echo "portSize: $portSize"
    echo "posFilaPal: $posFilaPal"
    echo "posColPal: $posColPal"
    echo "midaPaleta: $midaPaleta"
    echo "posFilaPil: $posFilaPil"
    echo "posColPil: $posColPil"
    echo "velFil: $velFil"
}

```

```

    echo "velCol: $velCol"
}
function comprobarValores
{
    echo "COMPROBAR VALORES"
    if [ $fil -lt $minFil ] || [ $fil -gt $maxFil ] ; then
        read -p "Files incorrectes, torna a provar (10-120): " fil
    fi

    if [ $cols -lt $minCol ] || [ $cols -gt $maxCol ] ; then
        read -p "Columnes incorrectes, torna a provar (10-36): " cols
    fi
    let maxPort=fil-1
    if [ $portSize -lt $minPort ] || [ $portSize -gt $maxPort ] ; then
        read -p "Porteria incorrecta, torna a provar (8-numFiles-1): " portSize
    fi
    if [ $posFilaPal -lt 2 ] || [ $posFilaPal -gt 118 ] ; then
        read -p "posFilaPal incorrecta, torna a provar (2-118): " posFilaPal
    fi
    if [ $posColPal -lt 2 ] || [ $posColPal -gt 35 ] ; then
        read -p "posColPal incorrecta, torna a provar (2-35): " posColPal
    fi
    if [ $midaPaleta -lt 3 ] || [ $midaPaleta -gt $maxPort ] ; then
        read -p "Mida paleta incorrecta, torna a provar (3-numFil-1): " midaPaleta
    fi
    if [ $posFilaPil -lt 2 ] || [ $posFilaPil -gt 118 ] ; then
        read -p "posFilaPil incorrecta, torna a provar (2-118): " posFilaPil
    fi
    if [ $posColPil -lt 2 ] || [ $posColPil -gt 35 ] ; then
        read -p "posColPil incorrecta, torna a provar (2-35): " posColPil
    fi

    if (( $(echo "${velFil} > 1.0 || ${velFil} < -1.0" | bc -l) )) ; then
        read -p "velFil incorrecta, torna a provar (-1, 1): " velFil
    fi
    if (( $(echo "${velCol} > 1.0 || ${velCol} < -1.0" | bc -l) )) ; then
        read -p "velCol incorrecta, torna a provar (-1, 1): " velCol
    fi
}

function comprobarfichero
{
    echo "COMPROBAR FICHERO"

    carlos='carlos'
    file='/home/milax/Documents/GitHub/FSOPrac1/params.txt'
    #file='/home/milax/FSOPrac1/params.txt'

    if [ -f $file ] && [ -e $file ]
    then
        num=0
    fi
}

```

```

while read line; do
if [ $num -eq 0 ]
then
    fil=`echo $line | cut -d' ' -f1`
    echo "$fil"
    cols=`echo $line | cut -d' ' -f2`
    echo "$cols"
    portSize=`echo $line | cut -d' ' -f3`
    echo "$portSize"
elif [ $num -eq 1 ]
then
    posFilaPal=`echo $line | cut -d' ' -f1`
    echo "$posFilaPal"
    posColPal=`echo $line | cut -d' ' -f2`
    echo "$posColPal"
    midaPaleta=`echo $line | cut -d' ' -f3`
    echo "$midaPaleta"

elif [ $num -eq 2 ]
then
    posFilaPil=`echo $line | cut -d' ' -f1`
    echo "$posFilaPil"
    posColPil=`echo $line | cut -d' ' -f2`
    echo "$posColPil"
    velFil=`echo $line | cut -d' ' -f3`
    echo "$velFil"
    velCol=`echo $line | cut -d' ' -f4`
    echo "$velCol"

elif [ $num -eq 3 ] && [ ! -z "$line" ]
then
    extraPilotes='true'
    posFilaPil=`echo $line | cut -d' ' -f1`
    echo "$posFilaPil"

    posColPil=`echo $line | cut -d' ' -f2`
    echo "$posColPil"
    velFil=`echo $line | cut -d' ' -f3`
    echo "$velFil"
    velCol=`echo $line | cut -d' ' -f4`
    echo "$velCol"
fi
let num=num+1

found='true'
done < $file
let maxPort=$fil-1

else
    echo "No existe el archivo $file, procedemos a crearlo"

```



```

        touch $file #si no existe el fichero lo creamos

    fi
}

function readFromKeyboard
{
    echo "READ FROM KEYBOARD"
    if [ -z $Nflag ] ; then

        read -p "Introduce el nombre del archivo que quieres usar: " file
    fi

#FILAS
    if [ -z $Fflag ] ; then
        read -p "Introduce el numero de filas: " fil
    fi
        let maxPort=fil-1
        #rows output
        truncate -s-1 file
        #echo -n "$fil " > $file
#COLUMNAS
    if [ -z $Cflag ] ; then
        read -p "Introduce el numero de columnas: " cols
    fi
        #columns output//lo guardamos en el fichero $file
        truncate -s-1 file
        #echo -n "$cols " >> $file

#PORTERIA
    if [ -z $Pflag ] ; then
        read -p "Introduce la medida de la porteria: " portSize
    fi
        #output porteria// lo guardamos en el fichero $file
        truncate -s-1 file

        #echo -n "$portSize" >> $file
        comprobarValores
    }

echo "BIENVENIDO/A AL PROGRAMA"
#primero de todo comprobamos que el fichero deseado existe
#si el fichero existe debemos comprobar que todos los parametros estan en regla
    #readFile
#en caso contrario pediremos los parametros que faltan porteculado al usuario
    #readFromKeyboard

while getopts 'n:f:c:p:m:0:1:' opcio; do
    case "${opcio}" in

```

```

n) echo "Option n has been chosen" ;Nflag='true';file="${OPTARG}";;
f) echo "Option f has been chosen";Fflag='true';fil=${OPTARG};;
c) echo "Option c has been chosen";Cflag='true';cols=${OPTARG};;
p) echo "Option p has been chosen";portSize=${OPTARG};Pflag='true';;
m) echo "Option m has been chosen";midaPaleta=${OPTARG};Mflag='true';;
0)IFS=',';read -a strarr <<<
"${OPTARG}";posFilaPal=${strarr[0]};posColPal=${strarr[1]};Oflag='true';;
1) IFS=',';read -a split <<< "${OPTARG}";posFilaPil=${split[0]};
    posColPil=${split[1]};velFil=${split[2]};velCol=${split[3]};OneFlag='true';;
    #*) error "Unexpected option ${opcio}" ;;
esac
done
shift $(( $OPTIND - 1 ))
echo "*****"
printf "encara ens falta tractar els següents elements: %s\n$* \n"
echo "*****";echo ""
comprobarFichero
comprobarValores
printVariables
writeFile

#readFromKeyboard
introducirRestantes

```

Conclusions:

Realitzant aquesta pràctica hem pogut entendre amb més precisió com funcionen els scripts i les comandes de la terminal. Durant la realització d'aquesta, ens hem trobat amb diferents problemes que, la majoria ja sabiem solucionar en altres llenguatges de programació, però havíem de descobrir com fer-ho en bash. Un d'ells ha sigut el fet de llegir i escriure fitxers de text. Un altre ha sigut separar els valors que llegim del fitxer. Ens va portar un temps també entendre com funcionava el "getops" i com utilitzar els flags al nostre favor per determinar quines variables faltaven per tractar. Coses que com bé hem comentat, sabíem fer en altres llenguatges però hem hagut d'aprendre en bash en aquesta ocasió. Per altre banda hem hagut de repasar conceptes de C ja que al no ser un llenguatge orientat a objectes les funcions i el main del programa conviuen en el mateix fitxer, cosa que fa que en codis mitjanament llargs sigui una mica embolicat tot.

Autoavaluació:

Concepte a avaluar	Puntuació
Decisions de disseny (bash i python). Ús de funcions amb paràmetres. Documentació.	0,75
Disseny del Joc de Proves (bash i python). Documentació	0,75
Creació d'un fitxer amb opcions vàlides passades per línia d'ordres amb una pilota	0,35
Importar opcions d'un fitxer i sobre escriure algunes opcions passades per la línia d'ordres (amb una pilota)	0,1
Demandar opcions concretes a l'usuari si no s'han entrat per línia d'ordres	0,4
Demandar opcions concretes a l'usuari si estan fora del rang	0,5
Passar més d'una pilota per línia d'ordres, controlant màxim 9 pilotes	0,2
Què es fa si s'afegeixen pilotes i el fitxer existeix?	0,25
Tractament d'errors/excepcions	0,25
Validació dinàmica dels rangs de les variables	0,5
Implementació i ús correcte de les comandes de la bash	1,8
Implementació i ús correcte de les comandes python	1,6

PRÀCTICA 1.2 : FONAMENTS DE SISTEMES OPERATIUS

Carlos Martínez i Genís Martínez

GEI
2021-2022

Pràctica avaluable



UNIVERSITAT
ROVIRA I VIRGILI

Índex

Índex.....	2
1.- Decissions de disseny	3
2.- Pla de proves	4
3.- Codi font, conclusions i autoavaluació.....	5

1.- *Decisions de disseny*

Per a realitzar aquesta part de la pràctica ens hem guiat bastant per la part 1.1 desenvolupada en bash. Hem utilitzat el disseny descendent i mètodes amb finalitats semblants als de bash ja que la finalitat de la pràctica és la mateixa però aplicada a dos llenguatges de programació diferents. Les funcions que hem desenvolupat aquest cop són:

- **introducirRestantes** - indica quins elements falten per introduir
- **printVariables** - llista els valors de les variables
- **comprobarValores** - comprova que els valors de les variables estiguin dins dels rangs correctes
- **comprobarFichero** - comprova que existeix el fitxer i trosseja les variables
- **readFromKeyboard** - segons les variables que el “getops” detecti que falten, les demana per teclat
- **main()** - programa principal on llegim i escrivim els fitxers, a part de cridar als altres mètodes

Com es pot apreciar, aquest cop no hem utilitzat mètodes específics per llegir i escriure els fitxers. Hem decidit fer-ho des de el programa principal ja que el fet de modificar les variables fora del main, no les mantenia modificades a l'hora d'escriure-les als fitxers. Tot i això vam investigar sobre el tema i vam descobrir que es podria arribar a fer de varies maneres ja sigui amb l'ús de diccionaris o amb el prefix “global” davant de certes variables per tractar-les com una variable global.

Un cop encapsulades les tasques en aquestes funcions, al mètode main(), com bé hem dit el que fem és gestionar l'arxiu d'entrada, cridar a les funcions pertinents mencionades abans i finalment escriure l'arxiu modificat i amb les dades correctes.

2.- Pla de proves

Per a comprovar que la nostra solució al problema és funcional, hem fet un joc de proves similar al de la fase 1.1, totes les proves les hem realitzat amb comandes de l'estil:

```
python PyPrac1.py -n nomFit -f 25 -c 40 -p 20 -o 22,30 -m 10 -1
21,20,-
1.0,0.3 ,4,5,-1.0,0.3
```

El que hem anat provant és:

- Hem anat canviant els valors de les variables per comprovar que es guardaven correctament.
- Hem creat fitxers amb posicions buides per veure com es comportava el programa.
- Hem creat fitxers amb variables fora de rang.
- Hem introduït per comanda variables fora de rang.
- Hem introduït per comanda un nom de fitxer inexistent.
- En cas de tornar a introduir variables errònies per segon cop, hem fet que el programa segueixi demanant-les en bucle fins que les introduïdes siguin correctes
- Hem provat que les excepcions estiguin ben tractades

Prova d'exemple:

```
milax@casa:~/Documents/GitHub/FS0Prac1$ python PyPrac1.py -n params312.txt -f 30 -c 20 -p 20 -o 30,22
None
Comencem a imprimir
params312.txt
Falten parametres a la primera linea:
Fil: 0, col: 0, portsize: 0

PRINT VARIABLES
files:0
columnes:0
portSize:0
posFilaPal:30
posColPal:30
midaPaleta:30
posFilaPil:0
posColPil:0
velFil:-2
velCol:-2
Introdueix les files de nou (10-120): 150
Introdueix les files de nou (10-120): 190
Introdueix les files de nou (10-120): 30
Introdueix les columnes de nou (10-36): 20
Introdueix la mida de la porteria de nou (8-numFil-1): 8
Introdueixi les files on hi ha pilota de nou (2-118): 20
Introdueixi es columnes on hi ha pilota de nou (2-35): 10
Introdueixi la velocitat x de la pilota de nou (-1, 1): 0.5
```


3.- Codi font, conclusions i autoavaluació

Codi font:

```
#!/usr/bin/env python3
import sys
import getopt
import os
from os.path import exists

fil=0
cols=0
portSize=0
f=None
minFil=10
maxFil=120

minCol=8
maxCol=36

minPort=8

minPort=8
temp=None
posFilaPal=None
posColPal=None
midaPaleta=0
posFilaPil=0
posColPil=0
velFil=-2
velCol=-2
prueba1=3

def
comprobarValores(fil,cols,portSize,posFilaPal,posColPal,midaPaleta,posFilaPil,posCo
lPil,velFil,velCol):
    contador=10
    if ((fil > 120 ) or (fil<10)):
        fil=input("Introdueix les files de nou (10-120): " )
        contador-=1
    if ((cols > 36) or (cols<10)):
        cols=input("Introdueix les columnes de nou (10-36): ")
        contador-=1
    maxPort=fil-1
    if((portSize>maxPort)or(portSize<8)):
        contador-=1
        portSize=input("Introdueix la mida de la porteria de nou (8-numFil-1): ")
    if ((posFilaPal<2) or (posFilaPal>118)):
        contador-=1
```



```

        posFilaPal=input("Introdueixi la fila on hi ha paleta de nou (2-118): ")
    if ((posColPal<2) or (posColPal>35)):
        contador-=1
        posColPal=input("Introdueixi la columna on hi ha paleta de nou (2-35): ")
    if ((midaPaleta<2) or (midaPaleta>118)):
        contador-=1
        midaPaleta=input("Introdueixi la mida de la paleta de nou (3-numFil-1): ")
    if ((posFilaPil<2) or (posFilaPil>118)):
        contador-=1
        posFilaPil=input("Introdueixi les files on hi ha pilota de nou (2-118): ")
    if ((posColPil<2) or (posColPil>35)):
        contador-=1
        posColPil=input("Introdueixi es columnes on hi ha pilota de nou (2-35): ")
    if ((velFil<-1.0) or (velFil>1.0)):
        contador-=1
        velFil=input("Introdueixi la velocitat x de la pilota de nou (-1, 1): ")
    if ((velCol<-1.0) or (velCol>1.0)):
        contador-=1
        velCol=input("Introdueixi la velocitat y de la pilota de nou (-1, 1): ")
    return contador

def
printVariables(fil,cols,portSize,posFilaPal,posColPal,midaPaleta,posFilaPil,posColPil,velFil,velCol):
    print("")
    print("PRINT VARIABLES")
    print("files:" + str(fil))
    print("columnes:" + str(cols))
    print("portSize:" + str(portSize))
    print("posFilaPal:" + str(posFilaPal))
    print("posColPal:" + str(posColPal))
    print("midaPaleta:" + str(midaPaleta))
    print("posFilaPil:" + str(posFilaPil))
    print("posColPil:" + str(posColPil))
    print("velFil:" + str(velFil))
    print("velCol:" + str(velCol))

def comprobarFichero(archivo):
    f=None
    print("COMPROBANDO FICHERO...")
    try:
        f = open(archivo)
        print("Hem pogut accedir al fitxer.")
        return True
    except (IOError, TypeError):
        print("No s'ha pogut obrir el fitxer.")
        return False
    #finally:
    #    return False
    #    f.close()

```

```

def readFromKeyboard(temp):
    print("READ FROM KEYBOARD")
    temp=temp+1
    print("temp vale "+(str(temp)))

def main():
    fil=0
    cols=0
    portSize=0
    midaPaleta=0
    posFilaPil=0
    posColPil=0
    velFil=-2
    velCol=-2
    prueba1=3
    firstCorrect=False
    Nflag=None
    Fflag=None
    Cflag=None
    Pflag=None
    Mflag=None
    Oflag=None
    Oneflag=None
    archivo=None
    posFilaPal=None
    args=sys.argv[1:]
    #arg_list1b=['-n','params312.txt','-f','30']
    try:
        opts,
args=getopt.getopt(args,"n:f:c:p:m:0:1:",["arxiu=", "files=", "columnes=", "porteria=",
,"mida=", "zero=", "one="])
    except getopt.GetoptError as error_message:
        print(error_message)
        #return False

    for opt,arg in opts:
        if opt in ['-n','--arxiu']:
            Nflag=True
            archivo=arg

        elif opt in ['-f','--files']:
            Fflag=True
            try:
                fil=int(arg)

            except:
                fil=int(input("Introdueix les files de nou: "))
        elif opt in ['-c','--columnes']:
            Cflag=True
            cols=int(arg)
        elif opt in ['-p','--porteria']:

```

```

        Pflag=True
        portSize=int(arg)
    elif opt in ['-m', '--mida']:
        Mflag=True
        midaPaleta=int(arg)
    elif opt in ['-0', '--zero']:
        Oflag=True
        posFilaPal=int(arg.split(",")[0])
        posColPal=int(arg.split(",")[1])
    elif opt in ['-1', '--one']:
        Oneflag=True
        try:
            posFilaPil=int(arg.split(",")[0])
            posColPil=int(arg.split(",")[1])
            velFil=float(arg.split(",")[2])
            velCol=float(arg.split(",")[3])
        except:
            print("Dades incorrectes")
print(Oneflag)
index=0
print("Comencem a imprimir")
try:
    print(archivo)
    f=open(archivo)
    with f:
        #with open(archivo,'r'):
            for line in f:
                if index==0:
                    line=line.replace('\n','')
                    string=line.split(" ")
                    if len(string)==3:
                        firstCorrect=True
                        fil=int(line.split(" ")[0])
                        cols=int(line.split(" ")[1])
                        portSize=int(line.split(" ")[2])
                    else:
                        print("Falten parametres a la primera linea:")
                        fil=0
                        cols=0
                        portSize=0
                    print("Fil: "+str(fil)+", col: "+str(cols)+", portsize:
"+str(portSize))

                if index==1:
                    line=line.replace('\n','')
                    string=line.split(" ")
                    if len(string)==3:
                        posFilaPal=int(line.split(" ")[0])
                        posColPal=int(line.split(" ")[1])
                        midaPaleta=int(line.split(" ")[2])
                if index ==2:

```

```

        line=line.replace('\n','')

        string=line.split(" ")

        if len(string)==4:
            posFilaPil=int(line.split(" ")[0])
            posColPil=int(line.split(" ")[1])
            velFil=float(line.split(" ")[2])

            velCol=float(line.split(" ")[3])
            index=index+1
except Exception as e:
    print(e)

printVariables(fil,cols,portSize,posFilaPal,posColPal,midaPaleta,posFilaPil,pos
ColPil,velFil,velCol)
contador=10
correct=0
AllCorrect=False
if fil==25:
    print("si")
while ((fil > 120 ) or (fil<10)):
    fil=input("Introdueix les files de nou (10-120): " )
    contador-=1
while ((cols > 36) or (cols<10)):
    cols=input("Introdueix les columnes de nou (10-36): ")
    contador-=1
maxPort=fil-1
while((portSize>maxPort)or(portSize<8)):
    contador-=1
    portSize=input("Introdueix la mida de la porteria de nou (8-numFil-1): ")
while ((posFilaPal<2) or (posFilaPal>118)):
    contador-=1
    posFilaPal=input("Introdueixi la fila on hi ha paleta de nou (2-118): ")
while ((posColPal<2) or (posColPal>35)):
    contador-=1
    posColPal=input("Introdueixi la columna on hi ha paleta de nou (2-35): ")
while ((midaPaleta<2) or (midaPaleta>118)):
    contador-=1
    midaPaleta=input("Introdueixi la mida de la paleta de nou (3-numFil-1): ")
while ((posFilaPil<2) or (posFilaPil>118)):
    contador-=1
    posFilaPil=input("Introdueixi les files on hi ha pilota de nou (2-118): ")
while ((posColPil<2) or (posColPil>35)):
    contador-=1
    posColPil=input("Introdueixi es columnes on hi ha pilota de nou (2-35): ")
while ((velFil<-1.0) or (velFil>1.0)):
    contador-=1
    velFil=input("Introdueixi la velocitat x de la pilota de nou (-1, 1): ")
while ((velCol<-1.0) or (velCol>1.0)):
    contador-=1

```

```

        velCol=input("Introdueixi la velocitat y de la pilota de nou (-1,1): ")

    printVariables(fil,cols,portSize,posFilaPal,posColPal,midaPaleta,posFilaPil,pos
ColPil,velFil,velCol)
    print("archivo: "+archivo)
    try:
        #if comprobarFichero(archivo):
        #    f=open(archivo,'w')
        #    f.close()
        #else:
            f=open(archivo, 'w')
            f.write(str(fil) + " " + str(cols) + " " + str(portSize) + '\n')
            f.write(str(posFilaPal)+" " + str(posColPal)+" "+str(midaPaleta)+'\n')
            f.write(str(posFilaPil)+" "+str(posColPil)+" "+str(velFil)+"
"+str(velCol)+'\n')
        except (TypeError,IOError):
            print("Hi ha hagut un error amb el fitxer")

if __name__ == "__main__":
    main()

```

Conclusions:

Realitzant aquesta pràctica hem pogut entendre amb més precisió com funciona el llenguatge python. Durant la realització d'aquesta, ens hem trobat amb diferents problemes que, la majoria ja sabem solucionar en altres llenguatges de programació, però havíem de descobrir com fer-ho en python. Un d'ells ha sigut el fet de llegir i escriure fitxers de text. Un altre ha sigut separar els valors que llegim del fitxer. Coses que com bé hem comentat, sabíem fer en altres llenguatges però hem hagut d'aprendre en python en aquesta ocasió. Per altre banda hem hagut de repasar conceptes de C ja que al no ser un llenguatge orientat a objectes les funcions i el main del programa conviuen en el mateix fitxer, cosa que fa que en codis mitjanament llargs sigui una mica embolicat tot.

Autoavaluació:

Concepte a avaluar	Puntuació
Decisions de disseny (bash i python). Ús de funcions amb paràmetres. Documentació.	0,75
Disseny del Joc de Proves (bash i python). Documentació	0,75
Creació d'un fitxer amb opcions vàlides passades per línia d'ordres amb una pilota	0,35
Importar opcions d'un fitxer i sobre escriure algunes opcions passades per la línia d'ordres (amb una pilota)	0,1
Demandar opcions concretes a l'usuari si no s'han entrat per línia d'ordres	0,4
Demandar opcions concretes a l'usuari si estan fora del rang	0,5
Passar més d'una pilota per línia d'ordres, controlant màxim 9 pilotes	0,2
Què es fa si s'afegeixen pilotes i el fitxer existeix?	0,25
Tractament d'errors/excepcions	0,25
Validació dinàmica dels rangs de les variables	0,5
Implementació i ús correcte de les comandes de la bash	1,8
Implementació i ús correcte de les comandes python	1,6

Mateixa autoavaluació per les dues fases (1.1 i 1.2).