



---

# 4ª FASE

---

Proyecto global



13 DE DICIEMBRE DE 2020

SERGI PELLEJERO DISHMEY

## Contenido

Documentació del Projecte Global .....	1
Manual de com jugar .....	1
Black Jack .....	3
SLOTS .....	4
Ruleta .....	5
Limitacions del joc.....	7
Dificultats trobades .....	7
Possibles millores, no implementades .....	7
Opinió personal, sobre el treball realitzat .....	7
Estructura del projecte .....	8
Arxius.....	10
Casino.py .....	10
Intro.py .....	13
Creditos.py.....	15
Slots.py .....	16
Ruleta.py.....	19
Blackjack.py .....	23
Slotsd.py .....	29
Ruletad.py.....	30
Cartas.py .....	31

## Documentació del Projecte Global

### Manual de com jugar

1. Primerament ens donarà la benvinguda, demanant-nos el nostre **nom** i el nostre **cognom**. Podem posar només el nom si volem.
2. Ens preguntarà com es pot referir-se a nosaltres i introduïrem una de les 5 opcions (Sr, Sra, Srita, Srito o Cosa). Podem introduir l'opció tant en minúscules com en majúscules.
3. Ens preguntarà si volem entrar al casino. Haurem d'introduir **"si"** o **"no"**, podrem introduir-lo tant en minúscules com en majúscules.
4. Seguidament ens trobarem un missatge que ens diria que per jugar necessitem pepoints i que haurem d'introduir una targeta de crèdit, ens preguntarà si estem d'acord amb això. Haurem d'introduir **"si"** o **"no"**, podrem introduir-lo tant en minúscules com en majúscules.
5. Ens preguntarà per un "numero de targeta de crèdit", "més de caducitat", "any de caducitat" i el "cvc". Només fa falta que introduïm una targeta de crèdit vàlida, no te perquè ser real. Exemple :
  1. Numero: 4000001234567899
  2. Més: 12
  3. Any: 2020
  4. CVC: 131
6. Seguidament ens mostrarà una taula de relació **EUROS/PEPEPOINTS**, podrem introduir un valor d'euros que sigui vàlid. Per a que sigui vàlid ha de sortir en la taula. Un cop introduïm la quantitat, ens preguntarà si volem comprar més, hem de tenir en conte que hi ha un límit de compra i son 100€, per així no fomentar la ludopatia. Quan preguntà si volem comprar més, haurem d'introduir **"si"** o **"no"**, podrem introduir-lo tant en minúscules com en majúscules.

7. Ens mostrarà un menú amb els jocs que hi ha en el casino:

1. Black jack
2. SLOTS
3. RULETA
0. Salir

Per Entrar a qualsevol dels jocs, només haurem d'introduir el numero del joc i pressionar enter.

## Black Jack

```
¡Bienvenido al Blackjack!
Usted tiene esta cantidad de PEPEPOINTS: 4000.
La cantidad de apuesta obligatoria minima de este juego es de 20.
Quiere empezar a jugar o salir al menu principal?
```

- Quan ens surti aquest missatge de benvinguda, haurem d'introduir “**jugar**” o “**salir**” per tornar al menú, ho podrem introduir tant en minúscula com en majúscula.
- Quan introduïm jugar, passarem al següent menú.

```
Esta es tu cantidad de pepepoints: 4000
1. Jugar
2. Cambiar Apuesta
0. Salir
Opcion:
```

- Tenim 3 opcions:
  - **Jugar:** Aquesta ens permet fer començar a introduir una aposta.
  - **Cambiar apuesta:** Aquesta opció ens permet canviar l'aposta per tirada per a que no sigui de 20 sinó més gran i així poder guanyar més.
  - **Salir:** Aquesta opció ens permet sortir al menú principal.

```
Cartas del crupier:
10 ♠ ♠
♠ ♠ ♠
♠ ♠ 10

P P
E E
P
E E
S S

Tus cartas:
3 ♠ ♠
3 ♠

3 ♥ ♥
3 ♥
(6)

Que te gustaria hacer?
1.Pedir carta
2.Quedarse
```

- Al començar a jugar ens mostrarà les cartes del crupier i les nostres cartes. Podrem seleccionar dos opcions:

1. **Pedir carta:** Si seleccionem “pedir una carta” ens donarà una altre, podrem demanar tantes com vulguem sempre que no ens passem de 21.
2. **Quedarse:** si seleccionem “quedarse” ens quedarem amb les cartes que tinguem i el crupier començarà a jugar, si guanyem tenint una puntuació més a prop de 21 que ell, guanyem el joc si ell té un numero mes a prop de 21 guanyarà ell.

## SLOTS

Bienvenido a la maquina tragaperras!  
 Usted tiene esta cantidad de PEPEPOINTS: 4000.  
 La cantidad de apuesta obligatoria de este juego es de 10.  
 Aqui puede encontrar la tabla de ganancias dependiendo de la cantidad que este apostando.

TABLA DE GANANCIA / APUESTA			
COMBINACION	APUESTA 10	APUESTA 25	APUESTA 50
PEPE PEPE PEPE	1500	3000	6000
7 7 7	500	1250	2500
BAR BAR BAR	200	500	1000
BELL BELL BELL	150	300	600
LIMON LIMON LIMON	90	150	300
NARANJA NARANJA NARANJA	60	100	200
CEREZA CEREZA CEREZA	30	50	100
PEPE * *	Premio igual a la combinacion de *		
PEPE PEPE *	Premio igual a la combinacion de *		

Quiere empezar a jugar o salir al menu principal? \_

- Quan ens surti aquest missatge de benvinguda, haurem d'introduir **"jugar"** o **"salir"** per tornar al menú, ho podrem introduir tant en minúscula com en majúscula.
- Quan introduïm jugar, passarem al següent menú.

```

Esta es la cantidad de puntos que tiene: 4000
1. Jugar
2. Mostrar tabla de ganancia
3. Cambiar apuesta
0. Salir
Opcion:
  
```

- Tenim 4 opcions:
  1. **Jugar:** Aquesta ens permet fer una tirada dels slots.
  2. **Mostrar tabla de ganancia:** Aquesta ens permet veure la taula que sortia a la benvinguda.
  3. **Cambiar apuesta:** Aquesta opció ens permet canviar l'aposta per tirada per a que no sigui de 10 sinó més gran i així poder guanyar més.
  4. **Salir:** Aquesta opció ens permet sortir al menú principal.

## Ruleta

¡Bienvenido a la ruleta!

Usted tiene esta cantidad de PEPEPOINTS: 4000.

La cantidad de apuesta obligatoria de este juego es de 20.

Aquí puede encontrar la tabla de apuesta.

	1ª DOCENA				2ª DOCENA				3ª DOCENA				
00	3	6	9	12	15	18	21	24	27	30	33	36	3ª fila
	2	5	8	11	14	17	20	23	26	29	32	35	2ª fila
0	1	4	7	10	13	16	19	22	25	28	31	34	1ª fila
	1-18			PAR							IMPAR	19-36	
	ROJO						NEGRO						

Quiere empezar a jugar o salir al menu principal?

- Quan ens surti aquest missatge de benvinguda, haurem d'introduir **"jugar"** o **"salir"** per tornar al menú, ho podem introduir tant en minúscula com en majúscula.
- Quan introduïm jugar, passarem al següent menú.

```
1. Jugar
2. Mostrar tabla de apuesta
0. Salir
Opcion:
```

- Tenim 3 opcions:
  1. **Jugar:** Aquesta ens permet fer començar a introduir una aposta.
  2. **Mostrar tabla de apuesta:** Aquesta ens permet veure la taula que sortia a la benvinguda.
  3. **Salir:** Aquesta opció ens permet sortir al menú principal.

```
Esta es la cantidad de puntos que tiene: 4000
Introduzca una apuesta.
(1a docena/ 2a docena/ 3a docena/ 1a fila/ 2a fila/ 3a fila/ numero/ color/ par / impar/ 1-18 / 19-36):
```

- Aquí podem introduir un valor de tots els que es veuen a la taula, ja sigui un numero, color o fila. Un cop hem introduït a que volem apostar ens demanarà quant volem apostar. Hem de tenir en compte que l'aposta mínim es 20 i podem fer totes les apostes que vulguem sempre que tinguem pepepoints.
- Quan ens preguntin que si volem seguir apostant haurem d'introduir **"si"** o **"no"**, podem introduir-lo tant en minúscules com en majúscules.
- Seguidament ens preguntarà si volem veure les nostres apostes, podem veure a que i quant hem apostat. Haurem d'introduir **"si"** o **"no"**, podem introduir-lo tant en minúscules com en majúscules.

```
Este es el numero en el que ha caido la bola 19.  
No has Ganado nada con tu apuesta: 1-18.  
Presiona enter para continuar...
```

- Si no hem guanyat res amb la nostra aposta, ens mostrarà un missatge amb l'aposta dient-nos que no hem guanyat res, en canvi si hem guanyat si que ens sortirà un missatge com aquest.

```
Este es el numero en el que ha caido la bola 9.  
Has ganado 40.0 con tu apuesta.  
Presiona enter para continuar...
```

- Depenent que quant hem apostat i a que tindrem un multiplicador més gran o més petit. Hem de tenir en compte que els decimals de pepepoints només es poden fer servir en la ruleta, quan sortim d'aquest joc els decimals desapareixen.



### Limitacions del joc

- Crec que arriba un punt en que es fa monòton, ja que només hi ha tres jocs, per tant trobo una limitació en quant aquest aspecte. També podria utilitzar els decimals de la ruleta en tot el casino per poder tenir més punts.
- Una altre limitació que trobo es que no es pot guardar la quantitat de pepepoints per tant sempre que surts de jugar perds el que has guanyat.
- El Black jack es una versió bàsica, ja que no es pot fer Split o doblar l'aposta.

### Dificultats trobades

- L'única dificultat trobada al fer el projecte va ser aconseguir que els pepepoints es poguessin utilitzar en tots els jocs, i no perdre els guanys.
- Això ho vaig arreglar passant-lo per paràmetres i retornant-lo.

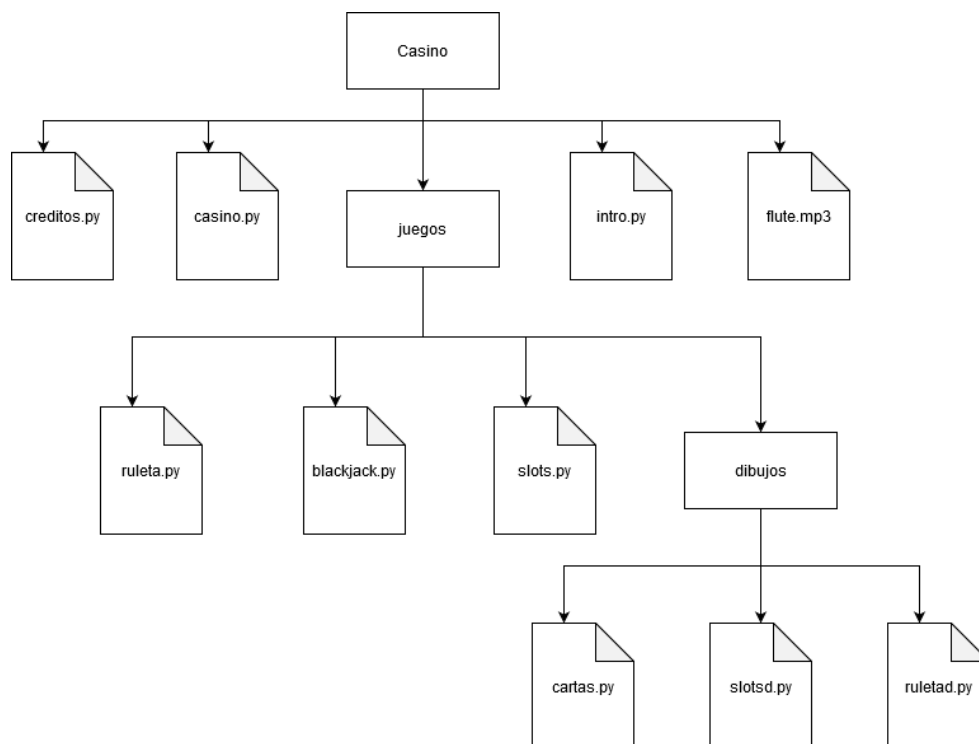
### Possibles millores, no implementades

- Una Possible millora seria fer el casino gràficament, ja que visualment seria més bonic però comporta una feineda per fer tot un programa tal com el que he fet gràficament.
- Una altra millora seria el poder guardar els pepepoints introduint un usuari i contrasenya en una base de dades o arxiu i així poder tenir sempre els pepepoints que has guanyat o perdut, en comptes d'haver de començar de nou.
- La utilització de decimals en tot el casino i no només en la ruleta.
- Un par més de jocs per abolir aquesta monotonia de només tres jocs.
- Al black jack afegiria que es pugui doblar l'aposta i que es pugui fer un Split.
- Utilitzaria menys globals i utilitzaria més retorns i paràmetres, per que no consumís tanta memòria.

### Opinió personal, sobre el treball realitzat

- En la meua opinió el treball que he realitzat es un molt bon treball tot i que es podria millorar i/o optimitzar molt més el codi. En general m'han anat en sortint fent el projecte, la veritat es que ho he gaudit ja que m'ha deixat ser creatiu i poder veure fins on puc arribar programant.

## Estructura del projecte



- L'estructura del projecte, comença amb la carpeta **"Casino"**, dins d'aquesta carpeta trobem, 3 arxius, un arxiu mp3 i un directori:
  - **creditos.py**: Dins d'aquest arxiu, trobem el codi del procés de compra dels pepepoints.
  - **casino.py**: Dins d'aquest arxiu, trobem el codi principal, on es criden els altres mòduls.
  - **intro.py**: Dins d'aquest arxiu trobem l'animació del inici del programa junt amb la reproducció de l'arxiu flute.mp3.
  - **flute.mp3**: Arxiu mp3 amb la musica que acompanya a l'animació de l'inici.
  - **juegos**: Directori que conte els jocs.
- Dins de **"juegos"**, trobem els jocs i un altre directori:
  - **ruleta.py**: Dins d'aquest arxiu, trobem el codi del joc de la ruleta.
  - **blackjack.py**: Dins d'aquest arxiu, trobem el codi del joc Black jack, que encara no esta acabat.
  - **slots.py**: Dins d'aquest arxiu, trobem el codi del joc slots.

- **dibujos:** Dins d'aquest directori, trobem els dibuixos que apareixen en els jocs.
- Dins de “**dibujos**”, trobem 3 arxius:
  - **cartas.py:** Dins d'aquest arxiu, trobem els dibuixos de les cartes, per al joc Black jack.
  - **slotsd.py:** Dins d'aquest arxiu, trobem els dibuixos dels slots, per al joc dels slots.
  - **ruletad.py:** Dins d'aquest arxiu, trobem els dibuixos per al joc de la ruleta.

## Arxius

### Casino.py

- **Casino.py** es l'arxiu principal del projecte, on es criden els jocs i l'animació del inici.
- Per a que funcioni correctament, importem els diferents mòduls:

```
import os, time, pycard
from intro import *
from juegos.blackjack import *
from juegos.slots import *
from credits import *
from juegos.ruleta import *
```

- Per utilitzar “**pycard**”, s'haurà d'instal·lar, mitjançant: “pip install <https://github.com/orokusaki/pycard/archive/master.zip>”.
- Dins d'aquest arxiu trobem les diferents funcions:
  - intro
  - processo
  - despedida
  - veri\_tarjeta
  - menu\_principal

### intro

```
def intro():
    global nom, trato, jugar
    animacion()
    nomape = input('BIENVENID@, a nuestro casino, "CASINOS PEPE", con quien tenemos el gusto de tratar?\nNombre y Apellido: ').
    nom = nomape[0]
    trato = input("Como deberiamos de referirnos a usted?\nComo Sr, Sra, Srita, Srito o Cosa?\n")
    if trato.lower() == 'cosa':
        jugar = input('%s quiere entrar a nuestro casino a jugar? ' %(trato.upper()[0]+trato[1:]))
        if jugar.lower() == "si":
            print("Procedamos a iniciar el proceso!")
            os.system("start https://ok.ru/video/205784418888")
            proceso()
            pepepoints = credits()
            menu_principal(pepepoints)
        else:
            despedida()
    elif trato.lower() == 'sr' or trato.lower() == 'sra' or trato.lower() == 'srita' or trato.lower() == 'srito':
        jugar = input('%s %s quiere entrar a nuestro casino a jugar? ' %(trato.upper()[0]+trato[1:], nom.upper()[0]+nom[1:]))
        if jugar.lower() == "si":
            print("Procedamos a iniciar el proceso!")
            proceso()
            pepepoints = credits()
            menu_principal(pepepoints)
        else:
            despedida()
```

- Aquesta funció es la principal de tot el programa la qual crida a les altres. Aquesta es la que executem en el codi principal. Dins d'aquesta trobem variables com:

- **nomape**: Aquesta variable, rep el nom i cognom del jugador.
- **nom**: Aquesta variable, agafa el nom de nomape.
- **trato**: Aquesta variable, rep un tracte per tractar al jugador com, sr, sra, srita, srito o cosa.
- **jugar**: Aquesta variable, rep si vol jugar al casino o no.
- Tambe trobem les funcions:
  - **despedida()** : on ens acomiadem del jugador.
  - **proceso()** : procés per que el jugador accepti l'acord per utilitzar la seva targeta de crèdit
  - **menu\_principal()** : menú principal on trobem els diferents jocs
- Dins trobem que depèn del tracte que escollim, si escollim el tracte de cosa, ens obrirà un vídeo a internet, el qual es la pel·lícula de “the thing” i si escollim un tracte normal, no ens apareix aquest vídeo.

### Proceso

```
def proceso():
    global terms
    terms = True
    while terms != False:
        os.system("cls")
        terms = input('Para poder proceder a jugar, necesitara "PEPEPOINTS".\nPara conseguir de estos, necesitara introducir su
        if terms.lower() == "si":
            veri_tarjeta()
            time.sleep(1)
            terms = False
        elif terms.lower() == "no":
            terms = False
            despedida()
        else:
            print("\nPerdone no le he entendido, puede volver a introducir si esta de acuerdo o no? ")
```

- Aquesta funció, conte la part del procés, per al qual s’ha d’acceptar per poder entrar al casino, i cridem a la funció “**veri\_tarjeta**”.

### Despedida

```
def despedida():
    print("\nRegresa cuando quieras.")
```

- Aquesta funció només imprimeix una frase de acomiadament.

### Veri\_tarjeta

```
def veri_tarjeta():
    global tarjeta
    tarjeta = False
    while tarjeta != True:
        os.system("cls")
        numero = input('Introduzca el numero de su tarjeta de credito: ')
        mes = int(input('Introduzca el mes de caducidad de su tarjeta de credito: '))
        ano = int(input('Introduzca el año de caducidad de su tarjeta de credito: '))
        cvc = int(input('Introduzca el cvc de su tarjeta de credito: '))
        tarjeta = pycard.Card(numero,mes,ano,cvc)
        if tarjeta.is_valid == True:
            print('\nLa tarjeta introducida es valida!')
            time.sleep(1.75)
            tarjeta = True
        else:
            print("\nLa tarjeta introducida no es valida, introduce una tarjeta valida!")
            time.sleep(1.75)
```

- Aquesta funció, la trobem dins de la funció proceso.
- veri\_tarjeta utilitza una llibreria especial anomenada “**pycard**”, la qual permet verificar targetes de crèdit.
- Si s’ha arribat aquest procés s’haurà d’introduir una targeta valida si o si, ja que sinó no sortirà del bucle.
- Per poder passar aquest procés, podem introduir una targeta amb un format vàlid. No fa falta que sigui real, només que tingui un format vàlid per exemple:
  - Numero: 4000001234567899
  - Mes: 12
  - Any: 2020
  - CVC: 131

### Menu\_principal

```
def menu_principal(puntos):
    os.system("cls")
    pepepoints = puntos
    accion = int(input("%s %s a que quiere jugar?\n\n1. Blackjack\n2. SLOTS\n3. Ruleta\n0. Salir\n Opcion: " %(trato.upper())[0]))
    while accion != 0:
        if accion == 1:
            print("blackjack")

        elif accion == 2:
            os.system("cls")
            pepepoints = juegoslot(pepepoints)

        elif accion == 3:
            os.system("cls")
            pepepoints = juegoruleta(pepepoints)

        accion = int(input("\n%s %s a que quiere jugar?\n\n1. Blackjack\n2. SLOTS\n3. Ruleta\n0. Salir\n Opcion: " %(trato.upper())[0]))
    despedida()
```

- Aquesta funció, es la segona mes important ja que es on un cop passat tot el procés trobem els diferents jocs del casino.

- `juegoblackjack()`
- `juegoslot()`
- `juegoruleta()`

## Intro.py

[illegible]

- Pip install playsound



### *animacion*

```
for x,y in d.items():  
    print(y)  
    time.sleep(0.10)  
    os.system("cls")
```

- Aquesta funció consta d'un diccionari amb la musica i les diferents escenes de l'animació, per poder mostrar tot això, hi ha un for que va recorrent el diccionari i imprimeix el dibuix, per cada escena hi ha un espai de temps de 0,1 segons i es neteja la pantalla amb el comanda del sistema.



## Creditos.py

- **Creditos.py** es on trobem la part per comprar els pepepoints.
- Dins d'aquest trobem 2 funcions:
  - `creditos`
  - `limitacion_dia`

*creditos*

```
def creditos():
    global eur, rep
    pepepoints = 0
    creditosdicc = {5 : 50, 10: 100, 15 : 225, 25 : 550, 35 : 900 , 45 : 1250, 50 : 2000}
    totalgastado = 0
    os.system("cls")
    print(" _____\n |TABLA DE VALOR PEPEPOINTS|\n | _____\n |EUROS | PEPEPOINTS|")
    rep = ''
    while rep.lower() != "no":
        eur = int(input("\n\nIntroduce una cantidad de euros: "))
        while eur != 5 and eur != 10 and eur != 15 and eur != 25 and eur != 35 and eur != 45 and eur != 50:
            print("La cantidad seleccionada no es valida, introduzca una cantidad valida")
            eur = int(input("\n\nIntroduce una cantidad de euros: "))
        totalgastado += eur
        if eur != 0:
            for x,y in creditosdicc.items():
                if eur == x:
                    pepepoints = pepepoints + int(y)
            print("Esta es la cantidad de PEPEPOINTS que tiene: %s ¤ " %(pepepoints))
            limitacion_dia(totalgastado)
        if rep.lower() == "no":
            print("Muchas Gracias por su compra!")
            time.sleep(2)
    os.system("cls")
    return pepepoints
```

- Aquesta funció es l'encarregada de permetre comprar els pepepoints.
- Mostra una taula amb el valor dels pepepoints respecte als euros.
- Funciona introduint una quantitat valida d'euros i sempre que no es superi el límit d'euros o no es vulgui comprar més, no sortirem del bucle.
- Es retorna els pepepoints per així poder utilitzar-los en els jocs.
- Trobem que es fa la crida a una funció anomenada limitacion dia.

limitacion dia

```
def limitacion_dia(dinero gastado):
    global eur, rep
    rep = input("\nTe gustaria comprar mas PEPEPOINTS? ")
    while rep.lower() != "si" and rep.lower() != "no":
        print("No ha introducido una opcion valida!")
        rep = input("\nTe gustaria comprar mas PEPEPOINTS? ")
    if rep.lower() == "si":
        if dinero gastado >= 100:
            print("Usted a superado la limitacion de dinero gastado por dia!")
            rep = "no"
            eur = 0
        elif rep.lower() == "no":
            time.sleep(0.1)
```

- Aquesta funció limita la despesa dels euros, per no fomentar la ludopatia, per això no es permet fer una despesa més gran de 100.

## Slots.py

- Dins de **slots.py**, trobem el codi sencer del joc slots, on es passa per paràmetre puntós, que equival als pepepoints per poder jugar.
- Principalment apareix una arracada del joc, per després donar-nos una benvinguda al joc. Ens mostra una taula amb les possibles combinacions i el seu benefici depenent de la aposta que s'hagi escollit. Per començar a jugar haurem d'introduir "**jugar**" o en el cas que vulguem tornar al menú principal haurem d'introduir "**salir**". Això ho podem introduir tant en minúscula com en majúscula.
- Un cop hem introduït "**jugar**", ens sortirà un menú amb 4 opcions:
  - 1. Jugar
  - 2. Mostrar tabla de ganancia
  - 3. Cambiar apuesta
  - 0. Salir
- Si juguem, apareixen els dibuixos de la maquina i depenen de si hem guanyat o no, ens mostrarà la quantitat de pepepoints que tenim, ja sigui menys o més.
- Podrem jugar sempre que tinguem pepepoints, en el moment en que no tinguem, no podrem jugar.
- Dins del codi, fem una crida a les següents llibreries i mòduls:
  - Os
  - Time
  - Random
  - Slotsd.py
- Trobem també 4 funcions:
  - menu
  - juegoslot
  - comprobacion
  - rueda

## Menu

```
def menu(puntos):
    global salir
    carga()
    print("\nBienvenido a la maquina tragaperras!\nUsted tiene esta cantidad de PEPEPOINTS: %s.\nLa cantidad de apuesta obligat
    tabla())
    salir = input("Quiere empezar a jugar o salir al menu principal? ")
    while salir != False and salir != True:
        if salir.lower() == "salir":
            salir = True
        elif salir.lower() == "jugar":
            salir = False
        else:
            salir = input("No ha respondido correctamente a la pregunta.\nVuelva a intentarlo: ")
```

- Aquesta funció, mostra el menú d'entrada del joc, on es mostra la benvinguda i si es vol jugar o sortir al menú principal.
- Fa una crida a la funció tabla que es troba en slotsd.py, per mostrar la taula de les combinacions.

## Juegoslot

```
def juegoslot(puntos):
    global salir, rueda1, rueda2, rueda3, apuesta
    pepepoints = puntos
    apuesta = 10
    menu(pepepoints)
    while pepepoints != 0 and salir != True:
        time.sleep(1)
        os.system("cls")
        accion = int(input("1. Jugar\n2. Mostrar tabla de ganancia\n3. Cambiar apuesta\n0. Salir\n Opcion: "))
        if accion == 1:
            pepepoints = pepepoints - apuesta
            rueda1 = rueda()
            rueda2 = rueda()
            rueda3 = rueda()
            pepepoints = comprobacion(pepepoints)
            paus = input("Presiona enter para continuar...")
        elif accion == 2:
            tabla()
            paus = input("Presiona enter para continuar...")
        elif accion == 3:
            apuesta = int(input("Introduzca la apuesta: "))
            while apuesta != 10 and apuesta != 25 and apuesta != 50:
                print("No ha introducido una apuesta valida!")
                apuesta = int(input("Introduzca la apuesta: "))
            elif accion == 0:
                salir = True
            else:
                print("No ha introducido una opcion valida, pruebe de nuevo...")
                time.sleep(1)
                print("Esta es la cantidad de puntos que tiene: %s P\n" %(pepepoints))
                os.system("cls")
    return pepepoints
```

- Aquesta funció es on trobem tot el joc.
- Consta d'un bucle del qual no es sortirà a no ser que "salir" sigui True o ens quedem sense "pepepoints". Ens mostrarà 4 opcions comentades anteriorment.

- La primera opció crida a la funció “**rueda**”, que aquesta retorna un valor i es guarda en aquestes 3 variables. Després es fa una comprovació de les 3 variables i el tipus d’aposta i depenen, es retorna un guany o una pèrdua.
- La segona opció ens torna a mostrar la taula de combinació més guanyys
- La tercera opció ens permet canviar l’aposta ja sigui (10, 25 o 50), sinó s’introdueix una quantitat correcta demanar que s’introdueixi una quantitat.
- La quarta opció o zero, canvia la variable salir a True, per poder sortir del joc
- Es retorna pepoints per poder utilitzar aquests en un altre joc si es vol.

### Comprobacion

```
def comprobacion(pepepoints):  
    print("%s %s %s" % (dslots(rueda1), dslots(rueda2), dslots(rueda3)))  
    if apuesta == 10:  
        if (rueda1 == "pepe" and rueda2 == "pepe" and rueda3 == "pepe"):  
            pepepoints = pepepoints + 1500  
        elif (rueda1 == "seven" and rueda2 == "seven" and rueda3 == "seven") or (rueda1 == "pepe" and rueda2 == "seven" and rueda3 == "seven"):  
            pepepoints = pepepoints + 500  
        elif (rueda1 == "bar" and rueda2 == "bar" and rueda3 == "bar") or (rueda1 == "pepe" and rueda2 == "bar" and rueda3 == "bar"):  
            pepepoints = pepepoints + 200  
        elif (rueda1 == "bell" and rueda2 == "bell" and rueda3 == "bell") or (rueda1 == "pepe" and rueda2 == "bell" and rueda3 == "bell"):  
            pepepoints = pepepoints + 150  
        elif (rueda1 == "limon" and rueda2 == "limon" and rueda3 == "limon") or (rueda1 == "pepe" and rueda2 == "limon" and rueda3 == "limon"):  
            pepepoints = pepepoints + 90  
        elif (rueda1 == "naranja" and rueda2 == "naranja" and rueda3 == "naranja") or (rueda1 == "pepe" and rueda2 == "naranja" and rueda3 == "naranja"):  
            pepepoints = pepepoints + 60  
        elif (rueda1 == "cereza" and rueda2 == "cereza" and rueda3 == "cereza") or (rueda1 == "pepe" and rueda2 == "cereza" and rueda3 == "cereza"):  
            pepepoints = pepepoints + 30
```

- Aquesta funció es l’encarregada de comprovar el resultat de les 3 variables que contenen el valor de la roda i depenent del resultat més el de l’aposta, es guanyarà una certa quantitat de pepoints o no, aquesta quantitat es retornada.

### Rueda

```
def rueda():  
    return random.choice(["seven", "bar", "bell", "limon", "naranja", "cereza", "pepe"])
```

- Aquesta funció es l’encarregada de retornar un valor aleatori de la llista que hi ha dins, així es defineixen les 3 variables denominades “**rueda**”, per després ser comprovades.

## Ruleta.py

- Dins de **ruleta.py**, trobem el codi sencer del joc de la ruleta, on es passa per paràmetre puntós, que equival als pepepoints per poder jugar.
- Principalment apareix una arracada del joc, per després donar-nos una benvinguda al joc. Ens mostra una taula amb les possibles apostes que és poden escollir. Per començar a jugar haurem d'introduir "**jugar**" o en el cas que vulguem tornar al menú principal haurem d'introduir "**salir**". Això ho podrem introduir tant en minúscula com en majúscula.
- Un cop hem introduït "**jugar**", ens sortirà un menú amb 3 opcions:
  - 1. Jugar
  - 2. Mostrar tabla de apuestas
  - 0. Salir
- Si juguem, ens mostra la quantitat de pepepoints que tenim i un seguit de possibles apostes que podem fer. Ens demanarà l'aposta i la quantitat que es vol apostar.
- Podrem jugar sempre que tinguem pepepoints, en el moment en que no tinguem, no podrem jugar.
- Dins del codi, fem una crida a les següents llibreries i mòduls:
  - os
  - time
  - random
  - ruletad.py
- Trobem també 4 funcions:
  - menu
  - juego
  - juegoruleta
  - comprobacion

## Menu

```
def menu(puntos):
    global salir
    carga()
    print("\n¡Bienvenido a la ruleta!\nUsted tiene esta cantidad de PEPEPOINTS: %s.\nLa cantidad de apuesta obligatoria de este\n\n")
    salir = input("Quiere empezar a jugar o salir al menu principal? ")
    while salir != False and salir != True:
        if salir.lower() == "salir":
            salir = True
        elif salir.lower() == "jugar":
            salir = False
        else:
            salir = input("No ha respondido correctamente a la pregunta.\nVuelva a intentarlo: ")
```

- Aquesta funció, mostra el menú d'entrada del joc, on es mostra la benvinguda i si es vol jugar o sortir al menú principal.
- Fa una crida a la funció tabla que es troba en ruletad.py, per mostrar la taula de les apostes.

## Juego

```
def juego(puntos):
    global listaApuesta
    pepepoints = puntos
    listaApuesta = {}
    segapu = ""
    while segapu.lower() != "no" and pepepoints != 0:
        os.system("cls")
        print("Esta es la cantidad de puntos que tiene: %s P\n" %(pepepoints))
        apostar = input("Introduzca una apuesta.\n(1a docena/ 2a docena/ 3a docena/ 1a fila/ 2a fila/ 3a fila/ numero/ color/ apuesta = int(input("Introduzca la apuesta, minimo 20: "))
        while (apuesta < 20) or (apuesta > pepepoints):
            apuesta = int(input("Introduzca una apuesta valida: "))
        listaApuesta[apostar] = apuesta
        pepepoints -= apuesta
        segapu = input("\n¿Quieres seguir apostando? ")
        segapu = input("\n¿Quiere ver su apuesta? ")
        while segapu.lower() != "no" and segapu.lower() != "si":
            segapu = input("\nIntroduzca una respuesta valida: ")
        if segapu.lower() == "si":
            for x,y in listaApuesta.items():
                print("\nApuesta: %s \nPuntos apostados: %s" %(x, y))
        pepepoints = comprobacion(pepepoints)
    return pepepoints
```

- Aquesta funció es l'encarregada de portar a terme el joc.
- Mentre la variable "segapu" sigui diferent a no o tinguem pepepoints, podem fer totes les apostes que vulguem. Consta de dues entrades, una per la aposta de la ruleta i l'altre la quantitat de pepepoints que es desitja apostar per aquella aposta.
- Es retorna els pepepoints per fer us d'aquets en altres parts del projecte.

## Juegoruleta

```
def juegoruleta(puntos):  
    global salir  
    pepepoints = puntos  
    menu(pepepoints)  
    while pepepoints != 0 and salir != True:  
        time.sleep(1)  
        os.system("cls")  
        accion = int(input("1. Jugar\n2. Mostrar tabla de apuesta\n0. Salir\n Opcion: "))  
        if accion == 1:  
            pepepoints = juego(pepepoints)  
            paus = input("Presiona enter para continuar...")  
        elif accion == 2:  
            tabla()  
            paus = input("Presiona enter para continuar...")  
        elif accion == 0:  
            salir = True  
        else:  
            print("No ha introducido una opcion valida, pruebe de nuevo...")  
    return pepepoints
```

- Aquesta funció es on trobem tot el joc.
- Consta d'un bucle del qual no es sortirà a no ser que “**salir**” sigui True o ens quedem sense “**pepepoints**”. Ens mostrarà 3 opcions comentades anteriorment.
  - La primera opció crida a la funció “**juego**”, que aquesta és on trobem la part per apostar.
  - La segona opció ens torna a mostrar la taula d'aposta de la ruleta
  - La tercera opció o zero, canvia la variable salir a True, per poder sortir del joc
- Es retorna pepepoints per poder utilitzar aquests en un altre joc si es vol.



### Comprobacion

```
def comprobacion(puntos):
    listaNumero = ["0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12", "13", "14", "15", "16", "17", "18", "19", "20", "21", "22", "23", "24", "25", "26", "27", "28", "29", "30", "31", "32", "33", "34", "35", "36"]
    listaFila = {"1a fila" : ["1", "4", "7", "10", "13", "16", "19", "22", "25", "28", "31", "34"], "2a fila" : ["2", "5", "8", "11", "14", "17", "20", "23", "26", "29", "32", "35"]}
    listaDoce = {"1a docena" : ["1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12"], "2a docena" : ["13", "14", "15", "16", "17", "18", "19", "20", "21", "22", "23", "24"], "3a docena" : ["25", "26", "27", "28", "29", "30", "31", "32", "33", "34", "35", "36"]}
    listaColor = {"rojo" : ["1", "3", "5", "7", "9", "12", "14", "16", "18", "19", "21", "23", "25", "27", "29", "31", "33", "35"], "verde" : ["2", "4", "6", "8", "10", "13", "15", "17", "20", "22", "24", "26", "28", "30", "32", "34", "36"]}
    listaMit = {"1-18" : ["1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12", "13", "14", "15", "16", "17", "18"], "19-36" : ["19", "20", "21", "22", "23", "24", "25", "26", "27", "28", "29", "30", "31", "32", "33", "34", "35", "36"]}
    listaPar = {"par" : ["2", "4", "6", "8", "10", "12", "14", "16", "18", "20", "22", "24", "26", "28", "30", "32", "34", "36"], "impar" : ["1", "3", "5", "7", "9", "11", "13", "15", "17", "19", "21", "23", "25", "27", "29", "31", "33", "35"]}
    pepepoints = puntos
    totalganado = 0
    time.sleep(2)
    os.system("cls")
    ruleta = random.choice(listaNumero)
    print("Este es el numero en el que ha caido la bola %s.\n" %(ruleta))
    for x,y in listaApuesta.items():
        if x in listaNumero:
            if x == ruleta:
                totalganado = totalganado + ( float(y) * 37)
                print("Has ganado %s con tu apuesta.\n" %(( float(y) * 37)))
            elif x in listaFila:
```

- Aquesta funció es l'encarregada de comprovar el resultat de la llista d'apostes que el jugador hagi fet i depenent del resultat més el de l'aposta, es guanyarà una certa quantitat de pepepoints o no, aquesta quantitat es retornada.



### Blackjack.py

- Dins de **blackjack.py**, trobem el codi sencer del joc del Black jack, on es passa per paràmetre puntos, que equival als pepepoints per poder jugar.
- Principalment apareix una arracada del joc, per després donar-nos una benvinguda al joc. Per començar a jugar haurem d'introduir "**jugar**" o en el cas que vulguem tornar al menú principal haurem d'introduir "**salir**". Això ho podrem introduir tant en minúscula com en majúscula.
- Un cop hem introduït "**jugar**", ens sortirà un menú amb 3 opcions:
  - 1. Jugar
  - 2. Cambiar Apuesta
  - 0. Salir
- Si juguem, ens mostra les nostres cartes i les cartes del crupier. Ens mostrarà dues opcions :
  - 1.Pedir carta
  - 2.Quedarse
- La primera opció ens donarà una altra carta i la segona opció ens permet mantenir-nos amb les cartes que tenim.
- Podrem jugar sempre que tinguem pepepoints, en el moment en que no tinguem, no podrem jugar.
- Dins del codi, fem una crida a les següents llibreries i mòduls:
  - os
  - time
  - random
  - cartas.py
- Trobem també 4 funcions:
  - preparacion
  - menu
  - juegoblack
  - calc
  - comprobacionjung
  - comprobacion
  - juego

### Preparacion

```
def preparacion():
    global cartas
    cartas = [
        '2♠', '3♠', '4♠', '5♠', '6♠', '7♠', '8♠', '9♠', '10♠', 'J♠', 'Q♠', 'K♠', 'A♠',
        '2♦', '3♦', '4♦', '5♦', '6♦', '7♦', '8♦', '9♦', '10♦', 'J♦', 'Q♦', 'K♦', 'A♦',
        '2♥', '3♥', '4♥', '5♥', '6♥', '7♥', '8♥', '9♥', '10♥', 'J♥', 'Q♥', 'K♥', 'A♥',
        '2♣', '3♣', '4♣', '5♣', '6♣', '7♣', '8♣', '9♣', '10♣', 'J♣', 'Q♣', 'K♣', 'A♣',
    ]
    random.shuffle(cartas)
    jugador.append(cartas.pop())
    crupier.append(cartas.pop())
    jugador.append(cartas.pop())
    crupier.append(cartas.pop())
```

- Aquesta funció es l'encarregada de escollir les dues cartes del començament tant del jugador com del crupier.
- Conte una llista amb totes les cartes.

### Menu

```
def menu(puntos):
    carga()
    print(f"¡Bienvenido al Blackjack!\nUsted tiene esta cantidad de PEPEPOINTS: {puntos},\nLa cantidad de apuesta obligatoria minima de este juego es de 20.\n" f"{puntos}")
    salir = input("Quiere empezar a jugar o salir al menu principal? ")
    while salir != False and salir != True:
        if salir.lower() == "salir":
            salir = True
        elif salir.lower() == "jugar":
            salir = False
        else:
            salir = input("No ha respondido correctamente a la pregunta.\nVuelva a intentarlo: ")
    return salir
```

- Aquesta funció, mostra el menú d'entrada del joc, on es mostra la benvinguda i si es vol jugar o sortir al menú principal.

### Juegoblack

```
def juegoblack(puntos):  
    global apuesta  
    pepepoints = int(puntos)  
    salir = menu(pepepoints)  
    apuesta = 20  
    while pepepoints != 0 and salir != True:  
        time.sleep(1)  
        os.system("cls")  
        print('Esta es tu cantidad de pepepoints: %s \n' %(pepepoints))  
        accion = int(input("1. Jugar\n2. Cambiar Apuesta\n0. Salir\n Opcion: "))  
        if accion == 1:  
            pepepoints = juego(pepepoints)  
            paus = input("Presiona enter para continuar...")  
        elif accion == 2:  
            apuesta = int(input('Introduce una apuesta: '))  
            while apuesta % 2 != 0 and apuesta < 20:  
                print('Introduce una apuesta valida: ' )  
                apuesta = int(input('Introduce una apuesta: '))  
            os.system('cls')  
        elif accion == 0:  
            salir = True  
        else:  
            print("No ha introducido una opcion valida, pruebe de nuevo...")  
    return pepepoints
```

- Aquesta funció es on trobem la crida al joc i el menú principal del joc.
- Consta d'un bucle del qual no es sortirà a no ser que “**salir**” sigui True o ens quedem sense “**pepepoints**”. Ens mostrarà 3 opcions comentades anteriorment.
  - La primera opció crida a la funció “**juego**”, que aquesta és on trobem la tot el joc.
  - La segona opció ens permet canviar l'aposta.
  - La tercera opció o zero, canvia la variable salir a True, per poder sortir del joc
- Es retorna pepepoints per poder utilitzar aquests en un altre joc si es vol.

## Calc

```
def calc(mano, quien):
    cartasEq = {
        '2A': 2, '3A': 3, '4A': 4, '5A': 5, '6A': 6, '7A': 7, '8A': 8, '9A': 9, '10A': 10, 'JA': 10, 'QA': 10, 'KA': 10, 'AA': [1, 11],
        '2B': 2, '3B': 3, '4B': 4, '5B': 5, '6B': 6, '7B': 7, '8B': 8, '9B': 9, '10B': 10, 'JB': 10, 'QB': 10, 'KB': 10, 'AB': [1, 11],
        '2V': 2, '3V': 3, '4V': 4, '5V': 5, '6V': 6, '7V': 7, '8V': 8, '9V': 9, '10V': 10, 'JV': 10, 'QV': 10, 'KV': 10, 'AV': [1, 11],
        '2D': 2, '3D': 3, '4D': 4, '5D': 5, '6D': 6, '7D': 7, '8D': 8, '9D': 9, '10D': 10, 'JD': 10, 'QD': 10, 'KD': 10, 'AD': [1, 11],
    }

    calc = 0
    if quien == 'jugador':
        if ((mano[0] == 'AA' or mano[0] == 'AB' or mano[0] == 'AV' or mano[0] == 'AD') and (mano[1] == '10A' or mano[1] == '10B' or mano[1] == '10V' or mano[1] == '10D') or
            (mano[0] == '10A' or mano[0] == '10B' or mano[0] == '10V' or mano[0] == '10D') and (mano[1] == 'JA' or mano[1] == 'JB' or mano[1] == 'JV' or mano[1] == 'JD')):
            calc += 21
        else:
            for x in mano:
                for z,k in cartasEq.items():
                    if z == x:
                        if x == 'AA' or x == 'AB' or x == 'AV' or x == 'AD':
                            pregu = int(input('Cuanto quieres que valga A? 1 o 11 : '))
                            while pregu != 1 and pregu != 11:
                                print('Introduce un valor correcto')
                                pregu = int(input('Cuanto quieres que valga A? 1 o 11 : '))
                            calc += pregu
                        else:
                            calc += k
            else:
                for x in mano:
                    for z,k in cartasEq.items():
                        if z == x:
                            if x == 'AA' or x == 'AB' or x == 'AV' or x == 'AD':
                                calc += 1
                            else:
                                calc += k
    return calc
```

- Aquesta funció és l'encarregada de portar a terme el càlcul de les baralles dels jugadors.
- Passem dos paràmetres, un es la llista on es tenen les cartes i el segon es a qui estem fent el càlcul.
- En el cas del jugador ens permet escollir entre 1 o 11 si ens surt un "A".

## Comprobacionjug

```
def comprobacionjug(valor):
    if valor > 21:
        print('Te has pasado, has perdido!')
        partida = False
        return partida
```

- Aquesta funció es l'encarregada de vigilar que si el jugador supera 21 s'acabi la partida retornant la variable "partida" en false.



### Comprobacion

```
def comprobacion(jug, crup, puntos):
    pepepoints = puntos
    if crup > 21:
        print('El crupier pierde, tu ganas!')
        pepepoints = pepepoints + (apuesta * 2)
    elif jug == crup:
        print('Empate, nadie gana')
        pepepoints = pepepoints + apuesta
    elif jug > crup:
        print('Has ganado al crupier, tu ganas!')
        pepepoints = pepepoints + (apuesta * 2)
    else:
        print('Tu pierdes :(')
    partida = False
    return [partida, pepepoints]
```

- Aquesta funció es l'encarregada de comparar qui està més a prop de 21 si el jugador o el crupier, si el jugador guanya, guanyarà la seva aposta per 2, si empaten es retorna el que ha apostat.
- Es retorna la variable "partida" i els "pepepoints" per acabar la partida.



## Juego

```
def juego(puntos):
    global crupier, jugador
    pepepoints = puntos
    pepepoints = pepepoints - apuesta
    primera_mano = True
    quedarse = False
    partida = True
    crupier = []
    jugador = []
    preparacion()
    crupierCart = dibujos(crupier)
    jugadorCart = dibujos(jugador)

    crupier_calc = calc(crupier, 'crupier')
    jugador_calc = calc(jugador, 'jugador')

    while pepepoints != 0 and partida != False:

        if quedarse == True:
            print('Cartas del crupier: \n{} ({}).format('\n'.join(crupierCart), crupier_calc))

            while crupier_calc <= 17:
                crupier.append(cartas.pop())
                crupierCart = dibujos(crupier)
                crupier_calc = calc(crupier, 'crupier')

            os.system('cls')
            print('Cartas del crupier: \n{} ({}).format('\n'.join(crupierCart), crupier_calc))
            x = comprobacion(jugador_calc, crupier_calc, pepepoints)
            partida = x[0]
            pepepoints = x[1]

        else:
            print('Cartas del crupier: \n{}'.format(crupierCart[0]+'\\n'+ cartalrev))
            print('\\nTus cartas:\\n{} ({}).format('\\n'.join(jugadorCart), jugador_calc))

            while quedarse != True and partida != False:
                opcion = int(input('\\nQue te gustaria hacer?\\n1.Pedir carta\\n2.Quedarse\\n'))
                if opcion == 1:
                    os.system('cls')
                    jugador.append(cartas.pop())
                    jugadorCart = dibujos(jugador)
                    jugador_calc = calc(jugador, 'jugador')
                    partida = comprobacionjug(jugador_calc)
                    print('Cartas del crupier: \n{}'.format(crupierCart[0]+'\\n'+ cartalrev))
                    print('\\nTus cartas:\\n{} ({}).format('\\n'.join(jugadorCart), jugador_calc))
                elif opcion == 2:
                    quedarse = True
                    os.system('cls')
            return pepepoints
```

- Aquesta funció es l'encarregada de portar a terme el joc.
- Mentre la variable "partida" sigui diferent a False o tinguem pepepoints, podem jugar tant com vulguem. Consta de dues opcions per demanar carta o quedar-se amb les que es té.
- Es retorna els pepepoints per fer us d'aquets en altres parts del projecte.

Slotsd.py

[illegible]

- Dins de **slotsd.py** trobem els dibuixos que apareixen en el joc slots.
- Consta de 3 funcions :
  - carga
  - tabla
  - dslots
- La funció **carga** mostra el falç carregament del joc més un dibuix d'una maquina traga perras.
- La funció **taula** mostra la taula de combinació més guanys del joc.
- La funció **dslots** obté per paràmetre la variable de la roda i mitjançant un diccionari i un for que recorre el diccionari, mostra el valor de la clau que te la variable.

```
def main():
    print("""
    TABLA DE GANANCIA / APUESTA

    COMBINACION | APUESTA 10 P | APUESTA 25 P | APUESTA 50 P

    PEPE PEPE PEPE | 1500 P | 3000 P | 6000 P
    7 7 7 | 500 P | 1250 P | 2500 P
    BAR BAR BAR | 200 P | 500 P | 1000 P
    BELL BELL BELL | 150 P | 300 P | 600 P
    LIMON LIMON LIMON | 90 P | 150 P | 300 P
    MANCITA MANCITA MANCITA | 60 P | 100 P | 200 P
    CEREZA CEREZA CEREZA | 30 P | 50 P | 100 P

    PEPE * * | Premio igual a la combinacion de *
    PEPE PEPE * | Premio igual a la combinacion de *
    """)

def display():
    global displaylots
    displaylots = [
        "seven",
        "bell",
        "limon",
        "manchita",
        "cerezita",
        "pepe"
    ]

    for x,y in displaylots.items():
        if x == y:
            print(y)
```

## Ruletad.py

[illegible]

- Dins de **ruletad.py** trobem els dibuixos que apareixen en el joc de la ruleta.
- Consta de 2 funcions:
  - **carga**
  - **tabla**
- La funció **carga** mostra el falç carregament del joc més un dibuix d'una maquina traga perras.
- La funció **taula** mostra la taula de les possibles apostes que es poden fer.



## Cartas.py

```
import time,os
cartalrev = "┌───┐\n P P \n EE \n P \n EE \n s s \n└───┘"

def carga():
    os.system("cls")
    print("""
      /\_/\
     (oo)\_____(
      (__)\       )\/\
         ||----w |
         ||     ||
    """)
    for x in range(0,101, 4):
        print("Cargando el juego",x,"% de 100 %")
        time.sleep(0.75)
    os.system("cls")

def dibujos(mano):
    global cartalrev
    list = []

    cartas = [
        "AA": "┌───┐\nA \n| \n| \n| \n| \n| \nA└───┘",
        "2A": "┌───┐\n2 \n| \n| \n| \n| \n| \n2└───┘",
        "3A": "┌───┐\n3 \n| \n| \n| \n| \n| \n3└───┘",
        "4A": "┌───┐\n4 \n| \n| \n| \n| \n| \n4└───┘",
        "5A": "┌───┐\n5 \n| \n| \n| \n| \n| \n5└───┘",
        "6A": "┌───┐\n6 \n| \n| \n| \n| \n| \n6└───┘",
        "7A": "┌───┐\n7 \n| \n| \n| \n| \n| \n7└───┘",
        "8A": "┌───┐\n8 \n| \n| \n| \n| \n| \n8└───┘",
        "9A": "┌───┐\n9 \n| \n| \n| \n| \n| \n9└───┘",
        "10A": "┌───┐\n10 \n| \n| \n| \n| \n| \n10└───┘",
        "JA": "┌───┐\nJ \n| \n| \n| \n| \n| \nJ└───┘",
        "QA": "┌───┐\nQ \n| \n| \n| \n| \n| \nQ└───┘",
        "KA": "┌───┐\nK \n| \n| \n| \n| \n| \nK└───┘",
    ]
```

- Dins de **cartes.py** trobem els dibuixos que apareixeran en el joc del Black jack.
- Consta de dos funcions:
  - carga
  - dibujos
- La funció **carga** mostra el falç carregament del joc més un dibuix d'una ma amb 4 cartes.
- La funció **dibujos**, té un diccionari amb els dibuixos de cada carta, aquests pasts a una llista i es retorna aquesta llista.

```
for x in mano:
    for z,y in cartas.items():
        if x == z:
            list.append(y)

return list
```