

Labs_p7

October 24, 2014



(C) Els professors de l'assignatura d'informàtica. Facultat de Física, Universitat de Barcelona

1 Sessió introductòria 7: control de flux

1.1 Objectiu

L'objectiu d'aquesta pràctica és la familiarització amb l'ús de les sentències de control de flux. Per practicar aquestes sentències dissenyarem un joc pas a pas, construint diverses funcions que finalment combinarem i que usaran `turtle` per a representar-lo gràficament.

Buscamines (simplificat): el joc es juga en una quadrícula de 10×10 . En dues de les cel·les hi ha amagades unes “mines”. En cada torn del joc es donen les coordenades d'una cel·la i l'ordinador indica el resultat de la jugada:

- Si en la cel·la hi ha una mina aquesta “explota” i el joc s'acaba
- Si no hi ha una mina es marca la cel·la amb un codi que indica quantes mines hi ha en les cel·les que l'envolten.

1.2 Pas 1: definició d'una funció d'entrada de dades

Un un programa necessiteu sovint demanar per consola un valor enter i que estigui entre dos límits: $\min \leq i \leq \max$. Implementeu una funció amb les especificacions següents de manera que es pugui usar per demanar el valor enter:

- Nom: `demana_enter()`
- Arguments: (text,min,max) amb valor per defecte $\min = 0$ i $\max = 9$
- Retorn: un valor enter $\min \leq i \leq \max$
- Funcionament: ha de demanar per teclat un valor enter donant el text que es rep com a argument. Si no verifica que $\min \leq i \leq \max$ s'ha de tornar a demanar. Quan es doni un valor en el rang acceptable l'ha de retornar

Comproveu en l'exemple que la funció respon correctament en tots els casos tenint en compte els valors per defecte. Recordeu també documentar-la usant “docstring”

1.3 Pas 2: inicialització de la finestra

Implementeu una funció que inicialitzi la part finestra gràfica (`turtle`) del joc:

- Nom: `ini_finestra()`
- Ha de crear una finestra de 600×600 píxels amb

- Color de fons al vostre gust
- Títol “Buscamines”
- Ha de retornar la finestra inicialitzada, que es podrà usar per a futures manipulacions

1.4 Pas 3: inicialització de la tortuga

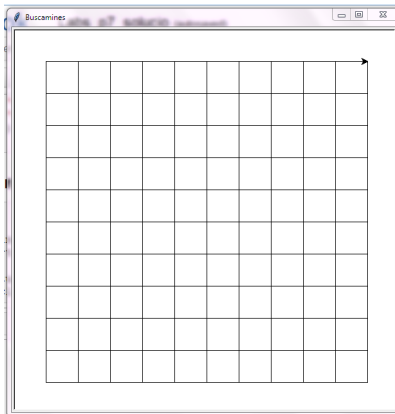
Implementeu una funció que inicialitzi la tortuga que usarem pel joc:

- Nom: `ini_tortuga()`
- Ha de crear la tortuga amb:
 - Gruix 1
 - Color negre
 - Velocitat 10
- Ha de retornar la tortuga creada, que s’usarà per traçar els gràfics

1.5 Pas 4: traçat de la graella (bucles for)

Implementeu una funció que traci una graella de 10×10 usant la tortuga creada anteriorment

- Nom: `ini_graella()`
- Arguments: (tortuga)

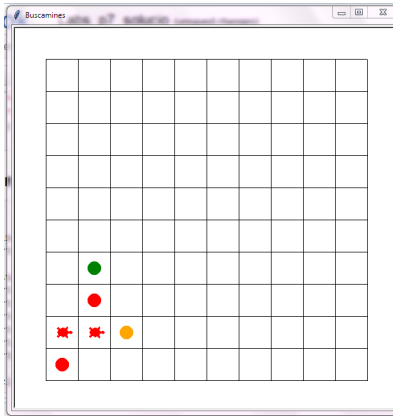


1.6 Pas 5: marca i control de jugada (sentència if)

Implementeu una funció que marqui el resultat d’una jugada, tenint en compte que el jugador indica les coordenades d’una cel·la donant fila i columna com dos números entre 0 i 9:

- Nom: `marca_jugada()`
- Arguments: (tortuga, Fg,Cg, F1,C1, F2,C2)
 - (Fg,Cg) coordenades de la cel·la donada pel jugador
 - (F1,C1) (F2,C2) coordenades de les mines
- Funcionament:
 - Marca una rodona verda a la cel·la i totes les del voltant estan buides
 - Marca una rodona taronja si la cel·la està buida pero hi ha una mina en alguna de les cel·les del voltant
 - Marca una rodona vermella si la cel·la està buida però hi ha dues mines en les cel·les dels voltants

- Marca una tortuga vermella si hi ha una mina a la cel·la seleccionada
- Retorn: **False** si la cel·la escollida pel jugador està buida i **True** si hi ha una mina



1.7 Pas 6: cos del programa (while)

Implementeu el cos del programa que controlarà el joc. Ha de començar per inicialitzar aleatòriament la posició de les dues mines. Després ha de constar d'un bucle **while** que s'executarà fins que es compleixi una de les dues condicions:

1. La funció `marca_jugada()` retorna **True**
2. S'han fet 98 jugades (vol dir que només queden les cel·les amb les dues mines)

A cada iteració del bucle el programa ha de:

1. Demanar les coordenades d'una cel·la
2. Comprovar la jugada

El programa ha d'acabar amb un missatge indicant al jugador si ha perdut o ha guanyat. Proveu el programa jugant una partida.

1.8 Millores opcionals

Podeu implementar les següents millores al joc.

1.8.1 1- Numeració de files i columnes

Amb Turtle podeu escriure un text a la pantalla usant la funció `tortuga.write("text")`. Useu-la per millorar la funció `ini_graella()` de manera que les files i columnes quedin numerades.

1.8.2 2- Més mines

Si sabeu usar llistes podeu modificar el programa per què el joc inclogui més de dues mines. En aquest cas cal usar una llista per contenir les coordenades de les mines i passar-la com a argument.

1.8.3 3- Ús del ratolí (*nivell avançat*)

Jugar entrant les coordenades per teclat es fa una mica pesat. Podem millorar el joc si fem que es pugui seleccionar una cel·la clicant sobre ella a la finestra. El programa següent crea una finestra i cada cop que es clica sobre ella imprimeix les coordenades del punt seleccionat:

```
In [5]: import turtle

        # Inicialització de la finestra
finestra = turtle.Screen()    # creem una nova finestra de dibuix
finestra.bgcolor("white")     # color verd clar
finestra.setup(600, 600)      # mida 350x500 punts
finestra.title("Llegir clics") # amb un títol

def fun_coord(Xr,Yr):
    print(Xr,Yr)

finestra.onscreenclick(fun_coord)
finestra.listen()
finestra.mainloop()

-137.0 82.0
-52.0 48.0
```

Proveu aquest programa i analitzeu-lo. Usant les funcions que hem definit abans, modifiqueu aquest codi de manera que el joc del buscamines s'executi usant el ratolí per seleccionar la cel·la.

Indicació: en aquest cas no cal que useu un `while`. El codi crida `fun_coord()` cada vegada que cliqueu el ratolí, de manera que podeu aprofitar-ho per cridar `marca_jugada()` i comprovar si s'ha clicat una mina. Per tancar la finestra des de dins de la funció crideu `finestra.exitonclick()`.

2 Exercici a entregar

Construiu un Notebook que resolgui, amb les explicacions i comentaris adequats per que s'entenguin els programes, els passos donats.

En acabar, entregueu el fitxer Notebook generat a través de [Campus Virtual](#)

In []: