



Road to 3BLD

Institut LLuís Companys

2n Batxillerat

Autor: Pol Sances Guirao

Tordera, Gener de 2024

Abstract

Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetur a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollicitudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maecenas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend consectetur. Nullam elementum, urna vel imperdiet sodales, elit ipsum pharetra ligula, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus.

Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetur a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollicitudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maecenas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend consectetur. Nullam elementum, urna vel imperdiet sodales, elit ipsum pharetra ligula, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus.

Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetur a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollicitudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maecenas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend consectetur. Nullam elementum, urna vel imperdiet sodales, elit ipsum pharetra ligula, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus.

Índex

Introducció	3
1 Marc Teòric	4
1.1 Història del Cub de Rubik	4
1.1.1 Invent i Introducció (1974-1980)	4
1.1.2 Primers Intents de Resolució (1980-1981):	4
1.1.3 Aparició dels Primers Campions (1982-1992):	4
1.1.4 L'Època dels Speedcubers (2003-2010):	4
1.1.5 L'Arribada de 3BLD (2003-2009):	4
1.1.6 Els Temps Cauen Dràsticament (2011-2020):	4
1.1.7 Avui en Dia (2023 en Endavant):	5
1.2 Entrant al concepte del Cub de Rubik	6
1.2.1 Interpretar el concepte del cub	6
1.2.2 Aplicar les matemàtiques al concepte	6
1.3 Notació dels Moviments	9
1.4 El Concepte de 3BLD	11
1.5 Fases de la Resolució	11
1.5.1 Memorització	11
1.5.2 Execució	14
2 Marc Pràctic	15
2.1 Elaboració d'una App amb python	15
2.2 Elaboració d'una web	23
2.2.1 L'Estructura	23
Referències	24

Introducció

Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetur a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollicitudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maecenas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend consectetur. Nullam elementum, urna vel imperdiet sodales, elit ipsum pharetra ligula, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus.

Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetur a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollicitudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maecenas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend consectetur. Nullam elementum, urna vel imperdiet sodales, elit ipsum pharetra ligula, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus.

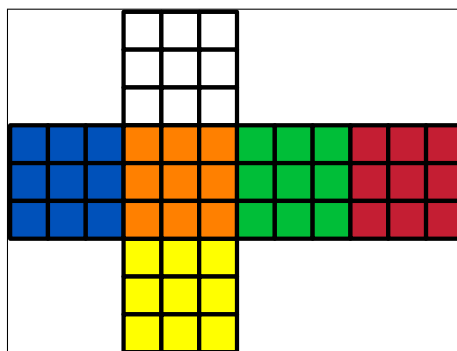


Figura 1: M2 U

i TOT ES VEU AL llibre [2]

1 Marc Teòric

1.1 Història del Cub de Rubik

1.1.1 Invent i Introducció (1974-1980)

El Cub de Rubik, concebut per Ernő Rubik el 1974 com una eina d'ensenyament per a conceptes espacials, es va llançar el 1975 a Budapest amb el nom "Cub Màgic". El seu disseny original constava de cares de colors sòlids. Aviat, aquest trencaclosques es va estendre per tot el món, però la seva resolució semblava un enigma sense solució clara.

1.1.2 Primers Intentos de Resolució (1980-1981):

David Singmaster, un estudiant d'enginyeria mecànica a Londres, va desenvolupar la primera notació per descriure els moviments del Cub i va crear el "mètode Singmaster" per resoldre'l, un fita important en la història del cub.

1.1.3 Aparició dels Primers Campions (1982-1992):

La dècada de 1980 va veure l'auge de la popularitat del Cub de Rubik, amb competicions de velocitat que van començar el 1982. Minh Thai es va convertir en el primer Campió Mundial. Els mètodes de resolució van evolucionar, i el mètode Friedrich de Jessica Fridrich es va convertir en un dels més populars.

1.1.4 L'Època dels Speedcubers (2003-2010):

La World Cube Association (WCA) es va fundar el 2003, establint estàndards i competicions oficials. Speedcubers com Feliks Zemdegs van dominar l'escena, establint rècords mundials en resolució estàndard i altres categories. La resolució del Cub es va convertir en una competició de temps.

1.1.5 L'Arribada de 3BLD (2003-2009):

La disciplina de resolució a cegues amb tres capes (3BLD) es va introduir en competicions, on els competidors havien de memoritzar i resoldre el cub sense veure'l. Tyson Mao es va convertir en el primer campió de 3BLD el 2003. Aquesta tècnica es va basar en memoritzar algorismes i seqüències de moviments.

1.1.6 Els Temps Cauen Dràsticament (2011-2020):

Amb l'expansió de comunitats en línia, la tècnica de 3BLD es va perfeccionar i els temps van millorar. Speedcubers com Max Park van establir rècords mundials impressionants en 3BLD. El 2017, Max Park

va aconseguir un rècord de 16,80 segons, demostrant el potencial d'aquesta disciplina.

1.1.7 Avui en Dia (2023 en Endavant):

El Cub de Rubik i el 3BLD continuen sent una font de passió i competició arreu del món. Les competències atrauen speedcubers de totes les edats i nivells d'habilitat, i la comunitat continua refinant les seves tècniques.

1.2 Entrant al concepte del Cub de Rubik

1.2.1 Interpretar el concepte del cub

Una gran majoria de la població ha tingut a les seves mans un cub de Rubik, i han intentat resoldre'l sense èxit. Això és totalment normal, ja que només el saben resoldre un 5,8% de les persones que ho han intentat [1].

Aquesta xifra es sol atribuir a la dificultat del cub, però després d'aprendre a fer el cub de rubik t'adones compte de què la raó no és aquesta. El fracàs a l'hora trobar la solució bé donat pel fet d'interpretar malament el concepte del funcionament del cub.

La majoria de les persones es pensa que el cub conté 54 "peces" de colors perquè calculen que per cada cara hi ha 9 peces i en un cub hi ha 6 cares, per tant estan treballant color a color.

$$\text{Nº Quadrats} = 3 \text{ Quadrats} * 3 \text{ Quadrats} * 6 \text{ Cares} = 54$$

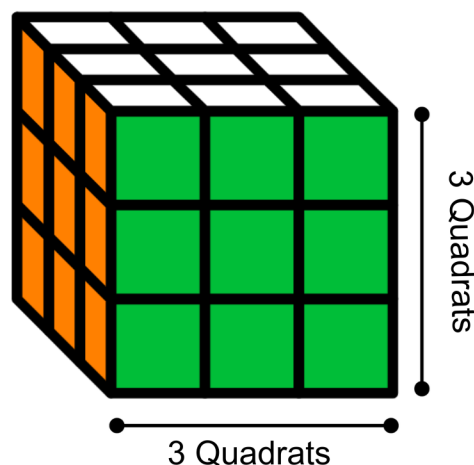


Figura 2: Plantejament típic però erroni del cub de rubik

La manera correcta d'interpretar el cub és pensar en el funcionament, com si el desmuntessis, ja que consta de 12 arestes i 8 cantonades, a més a més dels 6 centres que no poden permutar¹ amb cap altra peça ja que només roten.

1.2.2 Aplicar les matemàtiques al concepte

Després d'entendre el funcionament podem aplicar les matemàtiques i extreure el nombre de combinacions possibles del cub. En primer instant divid el càlcul el dos grups, per una part tenim cantonades i per l'altra arestes. Per la part de les cantonades es calcula:

¹Intercanvi de posició amb una altre peça i de l'ordre de tot el conjunt

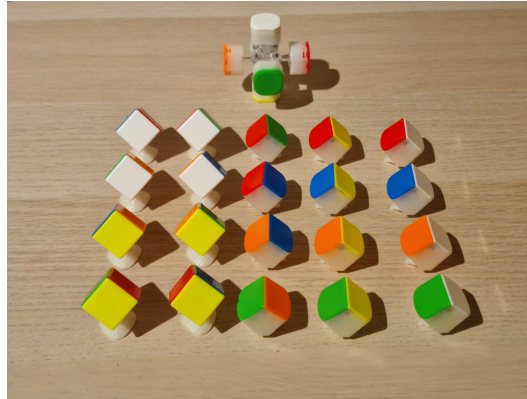


Figura 3: Cub Desmuntat

Tenim 8 cantonades que es poden posar de manera aleatoria en els 8 llocs, i això es calcula com a $8!^2$, després aquestes es poden orientar en 3 direccions diferents que matemàticament és 3^8 . Per tant les combinacions tèoriques possibles amb un cub de només cantonades són:

$$\text{Nº Combinacions cantonades tèoric} = 8! * 3^8 = 264.539.520$$

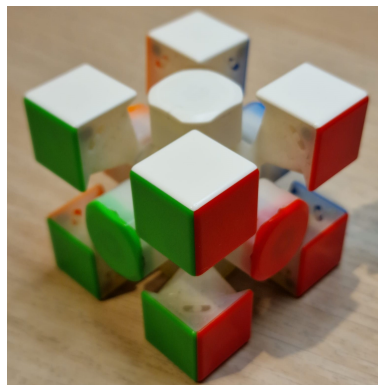


Figura 4: Cub amb només cantonades

Per altra banda tenim 12 arestes que igual que amb les cantonades es calcula com a $12!$ i com que les arestes del cub només tenen dos orientacions ho multiplicarem per 2^{12} . Per tant les combinacions tèoriques possibles amb un cub de només arestes són:

$$\text{Nº Combinacions arestes tèoric} = 12! * 2^{12} = 1.961.990.553.600$$

Llavors amb aquests càlculs podem extreure les conclusions que les combinacions possibles teòriques d'un cub de rubik són:

²! és el símbol de factorial o $8 * 7 * 6 * 5 * 4 * 3 * 2 * 1$

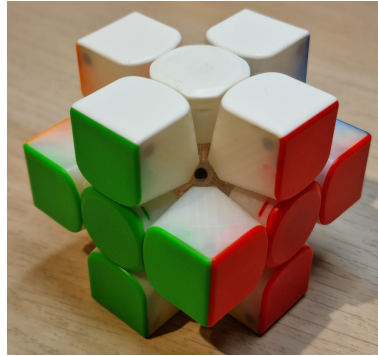


Figura 5: Cub amb només arestes

$$\text{N}^{\circ} \text{ Combinacions tèoriques} = 8! * 3^8 * 12! * 2^{12} = 519.024.039.293.878.272.000$$

Però cal dir que aquestes no són les combinacions totals reals del cub de rubik ja que aquestes combinacions estan calculades com si demuntessim el cub com a la 3 i recol·loquessim les peces en un estat aleatori. Llavors per calcular les reals s'han de dividir per 12 els casos per restriccions com les que es mostren en la següent figura.



Figura 6: Exemple de casos impossibles

I tot ben calculat queda:

$$\text{Nº Combinacions Reals} = \frac{8! * 3^8 * 12! * 2^{12}}{12} = 43.252.003.274.489.856.000$$

1.3 Notació dels Moviments

El cub de rubik es resol gràcies a identificar patrons i executar algorismes que resolen aquests patrons, aquests algorismes han d'estar escrits en alguna part per poder-los memoritzar i per això està la notació del cub de rubik.

La notació consta de 6 moviments (F,B,R,L,U,D), que correspon a (Front, Back, Right, Left, Up, Down) que son les respectives direccions en anglés. Per exemple si faig el moviment F gira la capa front la que eta més propera a mi en sentit horari, en canvi si fós F' seria antihorari. En les figures següents es mostra una representació gràfica per a cada capa.

És un concepte difícil d'entendre però de manera simplificada és girar la cara en sentit horari i anti-horari desde la cara que vulguis. En les figures següents es mostra una representació gràfica per a cada capa.



Figura 7: Exemples de Movimients F y F'

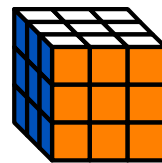


Figura 8: Exemples de Moviments B y B'

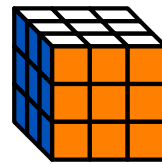


Figura 9: Exemples de Moviments R y R'



Figura 10: Exemples de Moviments L y L'

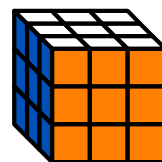
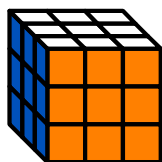


Figura 11: Exemples de Moviments U y U'

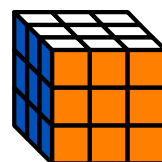
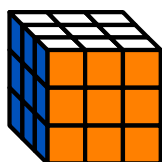


Figura 12: Exemples de Moviments D y D'

1.4 El Concepte de 3BLD

Per començar cal entendre el funcionament d'una resolució de blind, primer el cub és barrejat per una persona i el posa dins d'una capsa o un cube cover³, després es col·loca a la taula boca avall i la persona que l'ha de resoldre es pren el seu temps per respirar. Un cop fet això la persona que resol el cub encén el timer i destapa el cub, de manera que el temps comença a comptar i es comença a memoritzar. Un cop acabada la memorització el que resol el cub es tapa els ulls amb un antifàs i comença a resoldre el cub, mentre que una persona externa li posa una cartiluna entre el cub i la seva cara per evitar trampes i mirar per sota de l'antifàs. Tots aquests passos s'han d'executar perfectament per assegurar-se de la resolució compti.



Figura 13: Materials necessaris per poder executar blind

En la imatge anterior es poden veure el timer⁴, l'antifàs, la caixa per cobrir el cub, que en aquest cas jo utilizo una que tinc d'un cub, i a més a més uns cascos d'obra per aïllar-te del soroll ambient.

1.5 Fases de la Resolució

Com ja he esmentat a la secció anterior, completar el cub de Rubik amb els ulls tancats, es divideix en dos grans fases, memorització i execució. I dins d'aquestes fases hi han diferents procediments per poder aconseguir fer-ho correctament.

1.5.1 Memorització

Durant aquesta fase de memorització, com ja ho diu el seu nom, s'ha de memoritzar el cub. Molta gent pensa que els speedcubers que practiquem blind memoritzem el cub color per color mitjançant la

³Un cube cover és una tapa per cubs feta de cartró i que s'utilitza a les competicions

⁴El timer és el comptador amb la forma de les mans que es veu al centre de la imatge

			A	A	B							
			D		B							
			D	C	C							
E	E	F	I	I	J	M	M	N	Q	Q	R	
H		F	L		J	P		N	T		R	
H	G	G	L	K	K	P	O	O	T	S	S	
			U	U	V							
			X		V							
			X	W	W							

Pàg 12 / 24

mirant cap a tu i el centre blanc mirant cap a dalt.

BARREJA: F2 B R2 U'L2 U2 B' L' F2 U' B2 U L2 U R2 F2 L2 U' L2 F

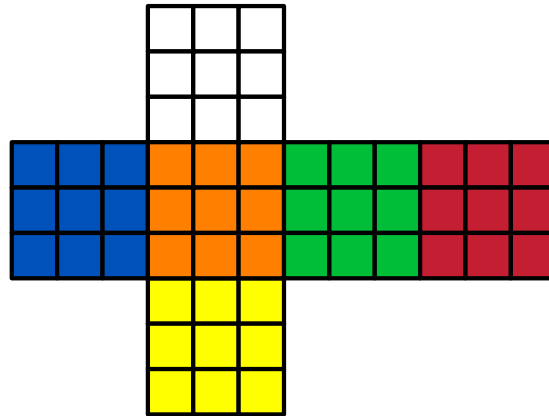


Figura 15: Cub barrejat per exemple de Blind

Començem mirant all lloc de U K com a l'esquema de lletres i veiem que és blanca a lloc U i taronja al lloc K, llavors ens fixem mirem l'esquema de lletres en l'aresta blanca-taronja i veiem que és la lletra D. Després ens fixem en el lloc de la lletra D i està una peça blanca-verda que és la lletra C, però és un cas especial, que més tard parlo a la secció d'execució però aquesta C es converteix en W, no s'ha de saber res més, només és per tenir el concepte entès. Això ho fem succesivament i obtenim que les lletres totals que ens hem de memoritzar són:

Memorització Arestes: DW LA BV PX RI GT N

(En aquest cas surten 13 perquè ha sigut una barreja amb cas especial)

Memorització Cantonades: U MH VS MC

(Igual que amb les arestes al ser un cas especial el valor es veu afectat)

Memorització Total DW LA BV PX RI GT NU MH VS MC

Lletres	Transcripció
DW	
LA	Los Ángeles
BV	
PX	
DF	

Lletres	Transcripció
DW	
LA	Los Ángeles
BV	
PX	
DF	

Lletres	Transcripció
DW	
LA	Los Ángeles
BV	
PX	
DF	

Lletres	Transcripció
DW	
LA	Los Ángeles
BV	
PX	
DF	

1.5.2 Execució

L'execució és la segona fase per completar el cub a cegues i es fa d'una manera molt diferent a la que es fa el cub de manera convencional. Quan tu resols el cub mirant tu fas rotacions al cub per buscar les peces i fer moviments, en canvi a les resolucions a cegues el cub no es rota en tota l'execució. Llavors les peces s'aconsegueixen col·locar a lloc gràcies a algorismes⁶.

Com que el cub no ha de rotar, les peces també s'han de quedar en el mateix lloc a l'acabar l'algoritme, és a dir, és executar un algoritme, mou les dues peces que vols i acabes amb la resta del cub igual però amb les dues peces canviades.

⁶És una seqüència de moviments que permet realitzar una tasca

2 Marc Pràctic

2.1 Elaboració d'una App amb python

Al saber tots aquests conceptes vaig decidir que hauria de portar aquesta eficiència a l'hora de resoldre, a l'aprenentatge. És a dir, que a la vegada que per resoldre el cub a cegues es busca l'eficiència en cada pas però a l'hora d'aprendre tota aquesta llista de conceptes és un procés molt lent. I és per això que per posar en pràctica el procés de memorització és a dir el traduir les lletres a les paraules per visualitzar imatges, el que vaig fer és una aplicació que et generi lletres, després te les amagui i tu hakis de posar les lletres que has memoritzat.

Aquesta app està realitzada amb el llenguatge de programació python, i amb la biblioteca tkinter per crear interfície gràfica⁷.

L'app consta de diferents labels⁸, botons i textboxes⁹, que es mostren o deixen de mostrar segons la funció que es doni i cada una d'aquestes funcions és escollida mitjançant els botons. Durant aquesta explicació del codi cal tenir en compte que les frases de color verd escrites després d'un estarisc són comentaris i no afecten al codi.

Per començar importem els paquets¹⁰ de tkinter i random, que el random es fa servir per generar les lletres aleatòriament. I després de tkinter importem els messageboxes.

```
1 import tkinter as tk
2 import random
3 from tkinter import messagebox
```

Listing 1: Importació de paquets

Després incio una classe on estarà tot el contingut i de l'app. Just a l'inici d'aquesta classe específic el títol de la finestra on es mostra l'app a més a més de les dimensions que tindrà i el logo que es mostrarà.

```
1 class MemorizeApp:
2     def __init__(self, root):
3         self.root = root
4         self.root.title("MemoApp by Pol Sances")
5         self.root.geometry("500x500")
```

⁷Una interfície gràfica és el que veus dins d'una aplicació, (els botons, menús...)

⁸És un text que l'usuari no pot alterar.

⁹És una "caixa" on l'usuari pot alterar el text.

¹⁰Un paquet és una col·lecció de fitxers i directoris necessaris per a una finalitat de programari


```
6 self.root.iconbitmap("Brain.ico")
```

Listing 2: Inici de la classe i especificació de la finestra

Just a sota del codi anterior declaro tots els objectes que estaran presents a l'app, labels que diuen el temps, textbox en la qual l'usuari omple amb un número i botons d'inci i tornar a inciar el procediment. Com a referència els objectes comencen amb un self. i la seva continuació és el nom d'objecte que és.

```
1 self.letters = self.generate_letters()
2
3 self.label = tk.Label(self.root, text="", font=("Arial", 24))
4 self.label.pack(pady=20)
5
6 self.time_entry = tk.Entry(self.root, font=("Arial", 14))
7 self.time_entry.insert(0, "Write_time_in_seconds")
8 self.time_entry.pack(pady=10)
9
10 self.start_button = tk.Button(self.root, text="Start", command=self
    .start_memorize)
11 self.start_button.pack(pady=5)
12
13 self.answer_entry = tk.Entry(self.root, font=("Arial", 14))
14 self.answer_entry.pack(pady=10)
15
16 self.submit_button = tk.Button(self.root, text="Submit", command=
    self.check_answers)
17 self.submit_button.pack(pady=5)
18
19 self.reset_button = tk.Button(self.root, text="Reset", command=self
    .reset)
20 self.reset_button.pack(pady=5)
21
22 self.answer_entry.config(state="disabled")
23 self.submit_button.config(state="disabled")
24 self.reset_button.config(state="disabled")
25
26 self.name = tk.Label(self.root, text="Made_by_Pol_Sances_Guirao",
    font=("Arial", 8))
27 self.name.pack(pady=20)
```

Listing 3: Declaració d'objectes necessaris pel funcionament de l'App

Després començo a escriure les funcions, en aquest cas començo amb la funció de generar les lletres gràcies al paquet random. Les funcions a python comencen amb el def i el nom de la funció, aquesta funció inicia una llista on s'escriuran les lletres generades. A sota on posa for i in range[20], significa que està creant un bucle 20 vegades on es genera una lletra i es s'escriu a la llista.

```
1 def generate_letters(self):
2     letters = []
3     for _ in range(20):
4         letter = chr(random.randint(65, 90))
5         letters.append(letter)
6     return letters
```

Listing 4: Funció per generar lletres

La següent funcions són les de començar a memoritzar, aquestes funció són les que s'atribueixen al botó d'inici de la memorització. El que fan es desactivar el botó start un cop clicat i afegir la llista de lletres dins del label perquè l'usuari pugui començar a memoritzar. Abans d'això la funció ha agafat el temps que havia introduït l'usuari i el converteix a milisegons perquè el programa mostri les lletres la quantitat de temps desitjada a més a més d'amagar les lletres a l'usuari.

```
1 def start_memorize(self):
2     self.start_button.config(state="disabled")
3     self.label.config(text="".join(self.letters))
4     time_delay = int(self.time_entry.get()) * 1000 # Convert user
5         seconds to milliseconds
6     self.root.after(time_delay, self.show_entry)
7     self.time_entry.pack_forget()
8
9 def show_entry(self):
10    self.label.config(text="")
11    self.answer_entry.config(state="normal")
12    self.submit_button.config(state="normal")
13    self.reset_button.config(state="normal")
14    self.answer_entry.focus_set()
```

Listing 5: Funcions pel botó d'inici

La següent funció és la de comprovar la resposta que has introduït, quan li dones al botó d'enviar la resposta, el que fa és dir-te si la resposta és correcta o incorrecta. I activar el botó de reset per poder

tornar a començar.

```

1      def check_answers(self):
2          user_answers = self.answer_entry.get().upper()
3          correct_answers = "".join(self.letters)
4
5          if user_answers == correct_answers:
6              messagebox.showinfo("Result", "Correct_answers!")
7          else:
8              messagebox.showinfo("Result", "Incorrect_answers.Try_again.")
9
10         self.reset()

```

Listing 6: Funció per comprovar la resposta

L'última funció és la de tornar començar, la qual et torna a generar les lletres i torna a començar el compte enrere, de manera resumida, torna a l'estat on li havies donat al botó d'inici.

```

1      def check_answers(self):
2          user_answers = self.answer_entry.get().upper()
3          correct_answers = "".join(self.letters)
4
5          if user_answers == correct_answers:
6              messagebox.showinfo("Result", "Correct_answers!")
7          else:
8              messagebox.showinfo("Result", "Incorrect_answers.Try_again.")
9
10         self.reset()

```

Listing 7: Funció per tornar a començar

Finalment tenim la part més important del codi que és el que crea la finestra principal i s'executa el bucle principal (mainloop) per mantenir l'aplicació en funcionament.

```

1  if __name__ == "__main__":
2      root = tk.Tk()
3      app = MemorizeApp(root)
4      root.mainloop()

```

Listing 8: Bucle per mantenir l'aplicació en funcionament

Llavors tot el codi junt queda:

```

1  # Importing packages

```

```
2
3 import tkinter as tk
4 import random
5 from tkinter import messagebox
6
7 # Begin and personalize window
8
9 class MemorizeApp:
10     def __init__(self, root):
11         self.root = root
12         self.root.title("Memo App by Pol Sances")
13         self.root.geometry("500x500")
14         self.root.iconbitmap("Brain.ico")
15
16     # Creating all the assets
17         self.letters = self.generate_letters()
18
19         self.label = tk.Label(self.root, text="", font=("Arial", 24))
20         self.label.pack(pady=20)
21
22         self.time_entry = tk.Entry(self.root, font=("Arial", 14))
23         self.time_entry.insert(0, "Write time in seconds")
24         self.time_entry.pack(pady=10)
25
26         self.start_button = tk.Button(self.root, text="Start", command=self
            .start_memorize)
27         self.start_button.pack(pady=5)
28
29         self.answer_entry = tk.Entry(self.root, font=("Arial", 14))
30         self.answer_entry.pack(pady=10)
31
32         self.submit_button = tk.Button(self.root, text="Submit", command=
            self.check_answers)
33         self.submit_button.pack(pady=5)
34
35         self.reset_button = tk.Button(self.root, text="Reset", command=self
            .reset)
```

```

36         self.reset_button.pack(pady=5)
37
38         self.answer_entry.config(state="disabled")
39         self.submit_button.config(state="disabled")
40         self.reset_button.config(state="disabled")
41
42         self.name = tk.Label(self.root, text="Made by Pol Sances Guirao",
43                               font=("Arial", 8))
44
45         self.name.pack(pady=20)
46
47         # Function to generate letters
48
49         def generate_letters(self):
50             letters = []
51             for _ in range(20):
52                 letter = chr(random.randint(65, 90))
53                 letters.append(letter)
54             return letters
55
56         # Function to show letters with the time that user put in the
57         time_entry textbox.
58
59         def start_memorize(self):
60             self.start_button.config(state="disabled")
61             self.label.config(text="".join(self.letters))
62             time_delay = int(self.time_entry.get()) * 1000 # Convert user
63             seconds to milliseconds
64             self.root.after(time_delay, self.show_entry)
65             self.time_entry.pack_forget()
66
67         # Function to hide letters
68
69         def show_entry(self):
70             self.label.config(text="")
71             self.answer_entry.config(state="normal")
72             self.submit_button.config(state="normal")
73             self.reset_button.config(state="normal")

```

```
70         self.answer_entry.focus_set()
71
72     # Function to check awnsers
73
74     def check_answers(self):
75         user_answers = self.answer_entry.get().upper()
76         correct_answers = "".join(self.letters)
77
78         if user_answers == correct_answers:
79             messagebox.showinfo("Result", "Correct_answers!")
80         else:
81             messagebox.showinfo("Result", "Incorrect_answers. Try again.")
82
83         self.reset()
84
85     # Reset function
86
87     def reset(self):
88         self.letters = self.generate_letters()
89         self.answer_entry.delete(0, tk.END)
90         self.label.config(text="")
91         self.start_button.config(state="normal")
92         self.answer_entry.config(state="disabled")
93         self.submit_button.config(state="disabled")
94         self.reset_button.config(state="disabled")
95
96
97 if __name__ == "__main__":
98     root = tk.Tk()
99     app = MemorizeApp(root)
100    root.mainloop()
```

Listing 9: Python example

El control de versions és pot veure a github a l'enllaç següent [Memo-App](#), i el resultat del codi en les diferents fases és el següent:

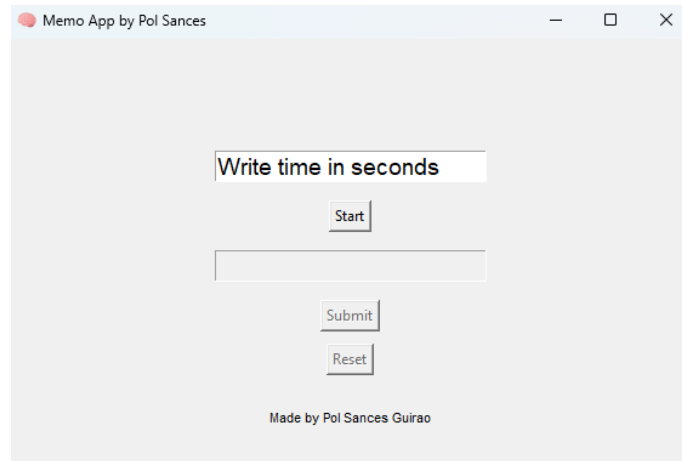


Figura 16: Fase Inicial

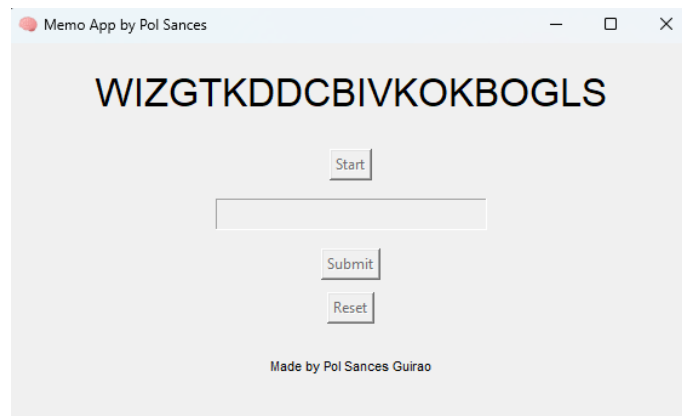


Figura 17: Fase de Memorització

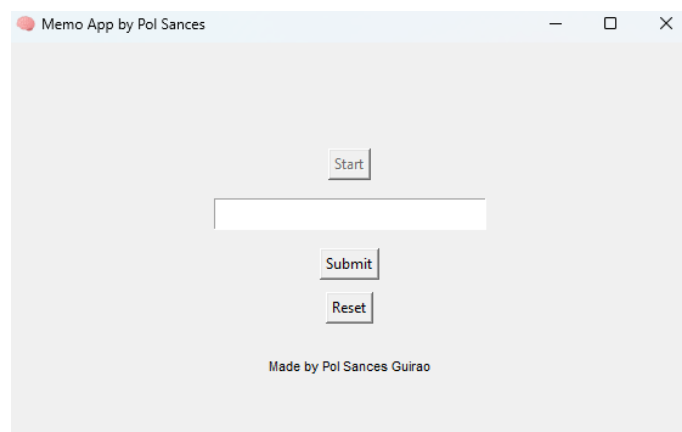


Figura 18: Fase d'introducció de la resposta

2.2 Elaboració d'una web

Per deixar-ho tot organitzat he decidit fer un web estàtica, és a dir que l'usuari no pot canviar el contingut de la web ni personalitzar-lo. La idea darrere de la web és fer com un tipus de central en la qual et porti als diferents continguts del treball, en resum està feta perquè serveixi com a tutorial a qualsevol que tingui una mica d'experiència en els cubs i es vulgui iniciar en el blind. Està redactada en HTML5 i CSS3, no és una web molt complexa però ha sigut desenvolupada sense gairebé experiència prèvia, només aprenent amb tutorials i llibres d'html.

2.2.1 L'Estructura

Com ja he dit està pensada per ser com una recepció que et dirigeix a les portes que vols, i té un estructura com la del diagrama següent:

Referències

[1] Redbull cub de rubik.

[2] Daniel Ross. *Rubik's Cube Best Algorithms*. Independently published, 2017.