

## Annexes :

Voici notre lien sur Github:

<https://github.com/lasource2019/Drumbox/blob/Drumbox-Version-Finale/Drumbox/DrumBox%20Version%20Finale>

Nous avons choisi la licence Creative Commons, car elle permet à l'auteur d'indiquer facilement les droits qu'il veut conserver et ceux auxquels ils renoncent afin de permettre à d'autres de réutiliser leur oeuvre.

Voici le code:

```
1 import pygame
2 from pygame.locals import *
3 import time
4 import numpy as np
5 import sounddevice as sd
6
7
8 #Initialisation de la drumbox et du clavier
9
10
11
12 #Création de la fenêtre graphique
13
14 res = (800, 540)
15 pygame.init()
16 pygame.display.set_caption("Drumbox")
17 pygame.mixer.init(14100, -16, 2, 2048)
18
19 #Mise en place du fond
20
21 fenetre_drumbox = pygame.display.set_mode(res)
22 drumbox_image = pygame.image.load("Images/DrumMachine.png")
23 fenetre_drumbox.blit(drumbox_image, [0, 0])
24 pygame.display.flip()
25
26 pygame.key.set_repeat(400, 30)
27
28 #Initialiation page d'accueil
29 testezDrum = pygame.image.load("Images/Testez.png").convert()
30 testezDrum = pygame.transform.scale(testezDrum, (230, 100))
31 creezMorceau = pygame.image.load("Images/créezvotremocreau.png").convert()
32 creezMorceau = pygame.transform.scale(creezMorceau, (230, 100))
33 retour = pygame.image.load("Images/retour.png")
34
35 #affichage page d'accueil
36
37 fenetre_drumbox.blit(retour, (720, 5))
38 fenetre_drumbox.blit(testezDrum, (130, 420))
39 fenetre_drumbox.blit(creezMorceau, (450, 420))
40 pygame.display.flip()
41
42 #Initialisation des sons
43
44 clap = pygame.mixer.Sound("Sons/Clap.wav")
45 kick = pygame.mixer.Sound("Sons/Kick.wav")
46 crash = pygame.mixer.Sound("Sons/crash.wav")
47 snare = pygame.mixer.Sound("Sons/Snare.wav")
48
49 #Intitialisation des images
50
51 ImgClap = pygame.image.load("Images/DrumMachine-Hat.png")
52 ImgKick = pygame.image.load("Images/DrumMachine-Kick.png")
53 ImgCrash = pygame.image.load("Images/DrumMachine-Crash.png")
54 ImgSnare = pygame.image.load("Images/DrumMachine-Snare.png")
55 ImgDrumbox = pygame.image.load("Images/DrumMachine.png")
```

## Drumbox (Léo, Jules, Adèle)

```
56
57 ImDo = pygame.image.load("Images/Clavier/ClavierDO.png")
58 ImRe = pygame.image.load("Images/Clavier/ClavierRE.png")
59 ImMi = pygame.image.load("Images/Clavier/ClavierMI.png")
60 ImFa = pygame.image.load("Images/Clavier/ClavierFA.png")
61 ImSol = pygame.image.load("Images/Clavier/ClavierSOL.png")
62 ImLa = pygame.image.load("Images/Clavier/ClavierLA.png")
63 ImSi = pygame.image.load("Images/Clavier/ClavierSI.png")
64
65 #Initialisation du clavier
66 clavier_image = pygame.image.load("Images/Clavier/Clavier.png")
67 x1 = 0
68 y1 = 0
69
70 black = (0, 0, 0)
71 pygame.display.flip()
72
73
74
75
76
77 #Initialisation du Séquenceur
78
79
80 #Initialisation des images
81
82 ImgKick_Sequencer = pygame.image.load("Images/Seq-Kick.png").convert_alpha()
83 ImgSnare_Sequencer = pygame.image.load("Images/Seq-Snare.png").convert_alpha()
84 ImgClap_Sequencer = pygame.image.load("Images/Seq-Clap.png").convert_alpha()
85 ImgHats_Sequencer = pygame.image.load("Images/Seq-Hats.png").convert_alpha()
86 ImgSilence_Sequencer = pygame.image.load("Images/Seq-Silence.png").convert_alpha()
87 img_fond = pygame.image.load("Images/FondPapier.png").convert_alpha()
88 sequenceur_img = pygame.image.load("Images/Fond.png").convert_alpha()
89
90
91 #Correspond à l'évènement de sélection d'image. Quand selected = 0, l'image n'est pas sélectionnée, quand selected = 1,
92 selected_kick = 0
93 selected_snare = 0
94 selected_clap = 0
95 selected_hats = 0
96 selected_silence = 0
97
98 #Positions des boutons des instruments
99 posx = [150, 250, 350, 450, 550]
100 posy = [35, 35, 35, 35, 35]
101
102
103
104
105 #Position des boutons des instruments lorsque la souris est relâchée
106 poscasex_kick = posx[0] - 35
107 poscasey_kick = posy[0] - 25
108 poscasex_snare = posx[1] - 35
109 poscasey_snare = posy[1] - 25
110 poscasex_clap = posx[2] - 35
111 poscasey_clap = posy[2] - 25
112 poscasex_hats = posx[3] - 35
113 poscasey_hats = posy[3] - 25
114 poscasex_silence = posx[4] - 35
115 poscasey_silence = posy[4] - 25
116
117 #Initialisation des sons du séquenceur
118 son = 0
119 kickson = 1
120 snareson = 2
121 clapson = 3
122 hatsson = 4
123 silenceson = 5
124 sequenceur = [0] * 14
125 kicksound = pygame.mixer.Sound("Sons/Kick.wav")
126 snaresound = pygame.mixer.Sound("Sons/Snare.wav")
127 clapsound = pygame.mixer.Sound("Sons/Clap.wav")
128 hatsound = pygame.mixer.Sound("Sons/crash.wav")
129
130 #Listes des coordonnées des cases du séquenceur
131 sequenceur_x = [125, 205, 285, 365, 445, 525, 605]
132 sequenceur_y = [80, 140]
133
```

## Drumbox (Léo, Jules, Adèle)

```
134 #Fonctions de la drumbox et du clavier
135
136 def SonClavierPressé():
137
138     if event.type == KEYDOWN:
139
140         if event.key == K_f:
141             clavier_image = ImDo
142             ChangementClavier(clavier_image)
143             frequency = 262
144             NoteClavier(frequency)
145
146         if event.type == KEYDOWN and event.key == K_g:
147             clavier_image = ImRe
148             ChangementClavier(clavier_image)
149             frequency = 293
150             NoteClavier(frequency)
151
152         if event.type == KEYDOWN and event.key == K_h:
153             clavier_image = ImMi
154             ChangementClavier(clavier_image)
155             frequency = 329
156             NoteClavier(frequency)
157
158         if event.type == KEYDOWN and event.key == K_j:
159             clavier_image = ImFa
160             ChangementClavier(clavier_image)
161             frequency = 349
162             NoteClavier(frequency)
163
164         if event.type == KEYDOWN and event.key == K_k:
165             clavier_image = ImSol
166             ChangementClavier(clavier_image)
167             frequency = 392
168             NoteClavier(frequency)
169
170         if event.type == KEYDOWN and event.key == K_l:
171             clavier_image = ImLa
172             ChangementClavier(clavier_image)
173             frequency = 440
174             NoteClavier(frequency)
175
176 def ChangementClavier(clavier_image):
177     fenetre_drumbox.blit(clavier_image, [0, 70])
178     pygame.display.flip()
179
180 def ChangementFond(drumbox_image):
181     fenetre_drumbox.blit(drumbox_image, [0, 0])
182     pygame.display.flip()
183
184 def RetourFond():
185     drumbox_image = pygame.image.load("Images/DrumMachine.png")
186     fenetre_drumbox.blit(drumbox_image, [0, 0])
187     pygame.display.flip()
188
```



## Drumbox (Léo, Jules, Adèle)

```
190 def NoteClavier(frequence):#fonction qui traite les données numériques pour créer un son. Variable : fréquence du son
191
192 # assignation de valeurs aux caractéristiques du signal analogique
193 amplitude = 10000
194 time = 1.0 # on définit le temps d'une note (1 seconde)
195
196 # élaboration du signal numérique
197 echantillonnage = np.arange(0, time, 1/44100) # définition de la fréquence d'échantillonnage
198 onde = amplitude * np.sin(2 * np.pi * frequence * echantillonnage)# création du signal numérique de formule : onde = ampli
199 tude*sin(2*pi*fréquence*échantillonnage)
200 ond_ond = np.array(onde, dtype=np.int16) # conversion en format onde que l'on code sur 16 bits
201 sd.play(ond_ond, blocking=True) # on convertit le signal numérique en son
202
203 def keydown(): #Fonction qui affecte des touches aux sons et au changement de fond quand la touche est pressée
204
205     if event.type == KEYDOWN:
206
207         if event.key == K_q:
208             kick.play()
209             drumbox_image = ImgKick
210             ChangementFond(drumbox_image)
211         if event.key == K_w:
212             snare.play()
213             drumbox_image = ImgSnare
214             ChangementFond(drumbox_image)
215         if event.key == K_a:
216             crash.play()
217             drumbox_image = ImgCrash
218             ChangementFond(drumbox_image)
219         if event.key == K_s:
220             clap.play()
221             drumbox_image = ImgClap
222             ChangementFond(drumbox_image)
223
224 def keyup(): #Fonction qui change le fond et arrête le son quand la touche est relâchée
225
226     if event.type == KEYUP:
227
228         if event.key == K_q:
229             kick.stop()
230             RetourFond()
231         if event.key == K_w:
232             snare.stop()
233             RetourFond()
234         if event.key == K_a:
235             crash.stop()
236             RetourFond()
237         if event.key == K_s:
238             clap.stop()
239             RetourFond()
240
241     pygame.display.flip()
242
243 def Batterie_clavier():
244     keydown()
245     keyup()
246     SonClavierPressé()
247
248 #Fonctions du séquenceur
249
250 def BoutonRelacher(position_x,position_y, instrument):
251     screen.blit(img_fond, (0,0))
252     screen.blit(sequenceur_img,(100, 70))
253     screen.blit(instrument,(position_x-35,position_y-25))
254     pygame.display.flip()
255     return
256
257
258
```

## Drumbox (Léo, Jules, Adèle)

```
259 #Fonction qui définit "l'alimentation" sur les cases du séquenceur
260 def Detection_Sequenceur(instrument_son, instrument, position_x_instrument, position_y_instrument):
261
262     for case in range(7):
263         if sequenceur_x[case] + 70 >= position_x_instrument >= sequenceur_x[case] and sequenceur_y[0] + 50 >= position_y_instrument >= sequenceur_y[0]:
264             Magnet(instrument, sequenceur_x[case], sequenceur_y[0])
265             sequenceur[case] = instrument_son
266     for case in range(7):
267         if sequenceur_x[case] + 70 >= position_x_instrument >= sequenceur_x[case] and sequenceur_y[1] + 50 >= position_y_instrument >= sequenceur_y[1]:
268             Magnet(instrument, sequenceur_x[case], sequenceur_y[1])
269             sequenceur[case + 6] = instrument_son
270
271 def Son(instrument):
272     son = instrument
273
274 def Magnet(instrument, case_x, case_y):
275     screen.blit(img_fond, (0,0))
276     screen.blit(sequenceur_img,(100, 70))
277     screen.blit(instrument,(case_x, case_y))
278     pygame.display.flip()
279
280 def SourisMouvement(instrument, instrument_position):
281     r = screen.blit(screen, instrument_position, instrument_position)
282     instrument_position.move_ip(event.rel) #a chaque fois il y a mouvement de souris, le personnage prend la nouvelle position du curseur
283     pygame.display.update((r,screen.blit(instrument, instrument_position)))
284
285 def ChangementImgSequenceur():
286     global poscase_x_kick, poscase_y_kick, poscase_x_snare, poscase_y_snare, poscase_x_clap, poscase_y_clap, poscase_x_hats, poscase_y_hats, poscase_x_silence, poscase_y_silence
287     screen.blit(img_fond, (0,0))
288     screen.blit(sequenceur_img,(100, 70))
289     screen.blit(ImgKick_Sequenceur, (poscase_x_kick, poscase_y_kick))
290     screen.blit(ImgSnare_Sequenceur, (poscase_x_snare, poscase_y_snare))
291     screen.blit(ImgClap_Sequenceur, (poscase_x_clap, poscase_y_clap))
292     screen.blit(ImgHats_Sequenceur, (poscase_x_hats, poscase_y_hats))
293     screen.blit(ImgSilence_Sequenceur, (poscase_x_silence, poscase_y_silence))
294     pygame.display.flip()
295
296     poscase_x_kick = posx[0] - 35
297     poscase_y_kick = posy[0] - 25
298     poscase_x_snare = posx[1] - 35
299     poscase_y_snare = posy[1] - 25
300     poscase_x_clap = posx[2] - 35
301     poscase_y_clap = posy[2] - 25
302     poscase_x_hats = posx[3] - 35
303     poscase_y_hats = posy[3] - 25
304     poscase_x_silence = posx[4] - 35
305     poscase_y_silence = posy[4] - 25
306
307 def SequenceurBoutonPressé():
308     #joue le son quand on appuie sur P
309     if event.type == KEYDOWN and event.key == K_p:
310         for loop in range(14):
311             if sequenceur[loop] == 1:
312                 kicksound.play()
313                 time.sleep(0.3)
314             if sequenceur[loop] == 2:
315                 snaresound.play()
316                 time.sleep(0.3)
317             if sequenceur[loop] == 3:
318                 clapsound.play()
319                 time.sleep(0.3)
320             if sequenceur[loop] == 4:
321                 hatssound.play()
322                 time.sleep(0.3)
323             if sequenceur[loop] == 5:
324                 time.sleep(0.5)
325
326 def MainSequenceur():
327     global selected_kick, selected_snare, selected_clap, selected_hats, selected_silence
328     ChangementImgSequenceur()
329     # boucle déplacement d'image pour le KICK
330     if event.type == pygame.MOUSEBUTTONDOWN and event.button == 1 and kick_position.collidepoint(event.pos):
331         selected_kick = 1
332
333     elif event.type == pygame.MOUSEBUTTONUP and event.button == 1 and selected_kick == 1:
334         posx[0],posy[0] = pygame.mouse.get_pos()
335         BoutonRelacher(posx[0],posy[0], ImgKick_Sequenceur)
336         selected_kick = 0
337         Detection_Sequenceur(kickson, ImgKick_Sequenceur, posx[0], posy[0])
338
```

## Drumbox (Léo, Jules, Adèle)

```
339     elif event.type == MOUSEMOTION and selected_kick == 1:
340         SourisMouvement(ImgKick_Sequanceur, kick_position)
341
342
343 # boucle déplacement d'image pour le SNARE
344     if event.type == pygame.MOUSEBUTTONDOWN and event.button == 1 and snare_position.collidepoint(event.pos):
345         selected_snare = 1
346
347     elif event.type == pygame.MOUSEBUTTONUP and event.button == 1 and selected_snare == 1:
348         posx[1],posy[1] = pygame.mouse.get_pos()
349         BoutonRelacher(posx[1],posy[1], ImgSnare_Sequanceur)
350         selected_snare = 0
351         Detection_Sequanceur(snareson, ImgSnare_Sequanceur, posx[1], posy[1])
352
353     elif event.type == MOUSEMOTION and selected_snare == 1:
354         SourisMouvement(ImgSnare_Sequanceur, snare_position)
355
356 # boucle déplacement d'image pour le CLAP
357     if event.type == pygame.MOUSEBUTTONDOWN and event.button == 1 and clap_position.collidepoint(event.pos):
358         selected_clap = 1
359
360     elif event.type == pygame.MOUSEBUTTONUP and event.button == 1 and selected_clap == 1:
361         posx[2],posy[2] = pygame.mouse.get_pos()
362         BoutonRelacher(posx[2],posy[2], ImgClap_Sequanceur)
363         selected_clap = 0
364         Detection_Sequanceur(clapson, ImgClap_Sequanceur, posx[2], posy[2])
365
366     elif event.type == MOUSEMOTION and selected_clap == 1:
367         SourisMouvement(ImgClap_Sequanceur, clap_position)
368
369 # boucle déplacement d'image pour les HATS
370     if event.type == pygame.MOUSEBUTTONDOWN and event.button == 1 and hats_position.collidepoint(event.pos):
371         selected_hats = 1
372
373     elif event.type == pygame.MOUSEBUTTONUP and event.button == 1 and selected_hats == 1:
374         posx[3],posy[3] = pygame.mouse.get_pos()
375         BoutonRelacher(posx[3],posy[3], ImgHats_Sequanceur)
376         selected_hats = 0
377         Detection_Sequanceur(hatsson, ImgHats_Sequanceur, posx[3], posy[3])
378
379     elif event.type == MOUSEMOTION and selected_hats == 1:
380         SourisMouvement(ImgHats_Sequanceur, hats_position)
381
382 # boucle déplacement d'image pour le SILENCE
383     if event.type == pygame.MOUSEBUTTONDOWN and event.button == 1 and silence_position.collidepoint(event.pos):
384         selected_silence = 1
385
386     elif event.type == pygame.MOUSEBUTTONUP and event.button == 1 and selected_silence == 1:
387         posx[4],posy[4] = pygame.mouse.get_pos()
388         BoutonRelacher(posx[4],posy[4], ImgSilence_Sequanceur)
389         selected_silence = 0
390         Detection_Sequanceur(silenceson, ImgSilence_Sequanceur, posx[4], posy[4])
391
392     elif event.type == MOUSEMOTION and selected_silence == 1:
393         SourisMouvement(ImgSilence_Sequanceur, silence_position)
394
395 SequenceurBoutonPressé()
396
397 continuer = 1
398 while continuer:
399     for event in pygame.event.get():
400
401         if event.type == QUIT:
402             continuer = 0
403
404         if event.type == MOUSEBUTTONDOWN and event.button == 1:
405             x1 = event.pos[0]
406             y1 = event.pos[1]
407
408
409 #affichage de la fenêtre de la drumbox quand est pressée la case "testez la batterie et le clavier")
410     if 130 <= x1 <= 360 and 420 <= y1 <= 520:
411         fenetre_drumbox = pygame.display.set_mode(res)#on ouvre une nouvelle fenêtre
412         drumbox_image = pygame.image.load("Images/DrumMachine.png")
413         fenetre_drumbox.blit(drumbox_image, [0, 0])
414         fenetre_drumbox.blit(clavier_image, [0, 70])
415         pygame.display.flip()
416         launched = True
417         while launched:
418             for event in pygame.event.get():
419                 if event.type == pygame.QUIT: #Fermeture de la fenêtre lorsque l'on appuie sur la croix rouge
420                     launched = False
421                     Batterie_clavier()
```



## Drumbox (Léo, Jules, Adèle)

```
422     elif 350 <= x1 <= 680 and 420 <= y1 <= 520:
423         res_sequenceur = (800, 250)
424         pygame.init()
425         pygame.display.set_caption("Sequenceur")
426         screen = pygame.display.set_mode(res_sequenceur)
427         kick_position = screen.blit(ImgKick_Sequenceur, (posx[0], posy[0]))
428         snare_position = screen.blit(ImgSnare_Sequenceur, (posx[1], posy[1]))
429         clap_position = screen.blit(ImgClap_Sequenceur, (posx[2], posy[2]))
430         hats_position = screen.blit(ImgHats_Sequenceur, (posx[3], posy[3]))
431         silence_position = screen.blit(ImgSilence_Sequenceur, (posx[4], posy[4]))
432         launched = True
433         while launched:
434             for event in pygame.event.get():
435                 if event.type == pygame.QUIT: #Fermeture de la fenetre lorsque l'on appuie sur la croix rouge
436                     launched = False
437                 MainSequenceur()
438     pygame.quit()
```

*Nous avons le droit d'utiliser les images et le son. Les images ont été dessinées par Léo grâce à logiciel Adobe Illustrator et les sons proviennent d'un logiciel payant, ce qui nous en confère la libre l'utilisation.*