

```

1  import random
2  import pygame
3  import pygame_gui
4
5  COULEUR = ['Spade', 'Heart', 'Clubs', 'Diamond']
6  VALEUR = { 2: 'Deux', 3: 'Trois', 4: 'Quatre', 5: 'Cinq', 6: 'Six', 7: 'Sept', 8: 'Huit', 9: 'Neuf', 10: 'Dix', 11: 'Valet', 12: 'Dame', 13: 'Roi', 14: 'As' }
7  Paths = [ '2S.jpg', '3S.jpg', '4S.jpg', '5S.jpg', '6S.jpg', '7S.jpg', '8S.jpg', '9S.jpg', '10S.jpg', 'JS.jpg', 'QS.jpg', 'KS.jpg', 'AS.jpg', '2H.jpg', '3H.jpg', '4H.jpg', '5H.jpg', '6H.jpg', '7H.jpg', '8H.jpg', '9H.jpg', '10H.jpg', 'JH.jpg', 'QH.jpg', 'KH.jpg', 'AH.jpg' ]
8  full_deck = []
9  for i in COULEUR:
10     for j in VALEUR.keys():
11         full_deck.append([j,i])
12  deck = full_deck
13  c = 0
14  for i in deck:
15     i.append(Paths[c])
16     c = c + 1
17  def color_value(card):
18     if card[1] == 'Spade':
19         return 1
20     elif card[1] == 'Heart':
21         return 2
22     elif card[1] == 'Clubs':
23         return 3
24     else:
25         return 4
26  class card(object):
27     def __init__(self, deck):
28         i = random.choice(range(len(deck)))
29         s = deck[i]
30         self.number = s[0]
31         self.color = s[1]
32         self.path = s[2]
33         deck.pop(i)
34  def distributing(full_deck):
35     deck = full_deck
36     deck1 = []
37     deck2 = []
38     counter = 0
39     while deck != []:
40         c = card(deck)
41         if counter % 2 == 0:
42             deck1.append([c.number,c.color,c.path])
43         else:
44             deck2.append([c.number,c.color,c.path])
45         counter = counter + 1
46     decks = [deck1 , deck2]
47     deck = full_deck
48     return decks
49  deck = full_deck
50  class player_human(object):
51     def __init__(self, name, my_deck):
52         self.name = name
53         self.deck = my_deck
54     def shuffle(self, my_deck):
55         new_deck = []
56         while my_deck != []:
57             c = card(my_deck)
58             new_deck.append([c.number, c.color,c.path])
59         self.deck = new_deck
60         return self.deck
61     def draw_card(self):
62         c = deck1[0]
63         print(str(self.name) + ' draws ' + str(c[0]) + ' of ' + str(c[1]))
64         deck1.pop(0)
65         return c
66     def call_to_end(self):
67         print(self.name + ' calls to end the game !!')
68     def quit(self):

```

```

69     print(self.name + 'quits!!')
70     return 0
71 def computer_strategy_1(deck2):
72     i = 0
73     j = 0
74     c = 0
75     new_deck = []
76     temp_deck = deck2
77     max = 2
78     max_index = 0
79     max_card = deck2[0]
80     while j < 26:
81         for i in temp_deck:
82             if i[0] > max:
83                 max = i[0]
84                 max_index = c
85                 max_card = i
86             c = c + 1
87         j = j + 1
88         new_deck.append([max_card[0], max_card[1], max_card[2]])
89         temp_deck.pop(max_index)
90         max_index = 0
91     return new_deck
92
93 decks = distributing(full_deck)
94 deck1 = decks[0]
95 deck2 = decks[1]
96 #deck2 = computer_strategy_1(deck2)
97 pygame.init()
98 images = [pygame.image.load("B.jpg")]
99 player1 = player_human('Player1', deck1)
100 clock = pygame.time.Clock()
101 time_delta = clock.tick(60) / 1000.0
102 pygame.display.set_caption('Bataille')
103 window_surface = pygame.display.set_mode((800, 600))
104 background = pygame.Surface((800, 600))
105 background.fill(pygame.Color('#ffb72e'))
106 manager = pygame_gui.UIManager((800, 600))
107 x = 800 // 2 - 50
108 y = 600 // 2 - 25
109 Start_button = pygame_gui.elements.UIButton(relative_rect=pygame.Rect((x, y - 75), (100, 50)), text='START',manager=manager)
110 Credit_button = pygame_gui.elements.UIButton(relative_rect=pygame.Rect((x, y), (100, 50)), text='Credits',manager=manager)
111 Quit_button = pygame_gui.elements.UIButton(relative_rect=pygame.Rect((x, y + 75), (100, 50)), text='Quit',manager=manager)
112 leave_button = 'l'
113 shuffle_button = 's'
114 draw_button = 'd'
115 play_button = 'p'
116 option_3_button = '3'
117 option_5_button = '5'
118 option_10_button = '10'
119 option_full_game_button = '13'
120 end_turn_button = 'e'
121 menu_button = 'm'
122 limit = 13
123 c1 = deck1[0]
124 c2 = deck2[0]
125 draw_allowed = True
126 shuffle_allowed = True
127 end_turn_allowed = False
128 Menu = pygame.display.get_surface()
129 e0 = 5.2
130 E = 0
131 CARDS_WON = 0
132 CARDS_LOST = 0
133 deck_img = pygame.image.load("JPEG\Green_back.jpg")
134 def game(manager,decks):
135     deck1 = decks[0]
136     arena = pygame.Rect((142.5, 200), (500, 200))
137     Player1_deck_surface = pygame.Rect((322.5, 420), (120, 168))
138     Player2_deck_surface = pygame.Rect((322.5, 10), (120, 168))
139     pygame.draw.rect(background, (75, 45, 5), arena)

```

```

140 pygame.draw.rect(background, (255, 255, 0), Player1_deck_surface)
141 pygame.draw.rect(background, (0, 255, 100), Player2_deck_surface)
142 manager.update(time_delta)
143 manager.update(time_delta)
144 deck_img = pygame.image.load("JPEG\Green_back.jpg")
145 e = 0
146 for i in deck1:
147     background.blit(deck_img, Player1_deck_surface)
148     background.blit(deck_img, Player2_deck_surface)
149     Player1_deck_surface = pygame.Rect((322.5 + e, 420 + e), (120, 168))
150     Player2_deck_surface = pygame.Rect((322.5 + e, 10 + e), (120, 168))
151     e = e + 0.2
152 manager.update(time_delta)
153 player1_score(0)
154 computer_score(0)
155 first_to(manager)
156 Game = pygame.display.get_surface()
157 return Game
158 def computer_draw(deck2):
159     c = deck2[0]
160     print('Computer' + ' draws ' + str(c[0]) + ' of ' + str(c[1]))
161     adresse = "JPEG/" + c[2]
162     deck2.pop(0)
163     cpath = pygame.image.load(adresse)
164     drawn_card_surface = pygame.Rect((420, 210), (120, 168))
165     deck_surface = pygame.Rect((322.5, 10), (126, 175))
166     pygame.draw.rect(background, pygame.Color('#ffb72e'), deck_surface)
167     e = 0
168     Player1_deck_surface = pygame.Rect((322.5, 10), (120, 168))
169     deck_img = pygame.image.load("JPEG\Green_back.jpg")
170     for i in deck1:
171         background.blit(deck_img, Player1_deck_surface)
172         Player1_deck_surface = pygame.Rect((322.5 + e, 10 + e), (120, 168))
173         e = e + 0.2
174     pygame.time.delay(100)
175     background.blit(cpath, drawn_card_surface)
176     return c
177 def first_to(manager):
178     f = pygame.font.Font('freesansbold.ttf', 25)
179     txt1 = f.render(str('First to :'), 1, (225, 225, 225), '#4b2d05')
180     txtrect1 = pygame.rect.Rect((350, 225), (100, 25))
181     pygame.draw.rect(background, (75, 45, 5), txtrect1)
182     background.blit(txt1, txtrect1)
183 def end(manager):
184     background = pygame.Surface((800, 600))
185     background.fill(pygame.Color('#ffb72e'))
186     manager.update(time_delta)
187     manager.clear_and_reset()
188 def see_first_card(deck1):
189     c1 = deck1[0]
190     adresse = "JPEG/" + c1[2]
191     cpath = pygame.image.load(adresse)
192     first_card_surface = pygame.Rect((322.5, 420), (126, 175))
193     background.blit(cpath, first_card_surface)
194 def battle(card1, card2):
195     if card1[0] > card2[0]:
196         return 1
197     elif color_value(card1) > color_value(card2) and card1[0] == card2[0]:
198         return 1
199     else:
200         return 0
201 def won(e, deck_img):
202     card_won_surface = pygame.Rect((12.5 + e, 210 + e), (120, 168))
203     background.blit(deck_img, card_won_surface)
204 def lost(e, deck_img):
205     card_won_surface = pygame.Rect((660 + e, 210 + e), (120, 168))
206     background.blit(deck_img, card_won_surface)
207 def player1_score(score):
208     f = pygame.font.Font('freesansbold.ttf', 25)
209     txt = f.render(str(score), 1, (0, 50, 0), '#ffb72e')
210     txtrect = pygame.rect.Rect((50, 399), (20, 20))

```

```

211 pygame.draw.rect(background, pygame.Color('#ffb72e'),txtrect)
212 background.blit(txt, txtrect)
213 def computer_score(score):
214     f = pygame.font.Font('freesansbold.ttf', 25)
215     txt = f.render(str(score), 1, (0, 50, 0), '#ffb72e')
216     txtrect = pygame.rect.Rect((725, 390), (20, 20))
217     pygame.draw.rect(background, pygame.Color('#ffb72e'), txtrect)
218     background.blit(txt, txtrect)
219 def clear_arena():
220     arena = pygame.Rect((142.5, 200), (500, 200))
221     pygame.draw.rect(background, (75, 45, 5), arena)
222     pygame.time.delay(100)
223 def winner(manager):
224     print("Player1 is the WINNER!!!")
225     f = pygame.font.Font('freesansbold.ttf', 76)
226     txt1 = f.render(str('CONGRATULATIONS'), 1, (250, 0, 25), 'ffffff')
227     txtrect1 = pygame.rect.Rect((0, 245), (800, 80))
228     pygame.draw.rect(background, (255,255,255), txtrect1)
229     background.blit(txt1, txtrect1)
230     g = pygame.font.Font('freesansbold.ttf', 50)
231     txt2 = g.render(str('YOU WON'), 1, (250, 0, 25), 'ffffff')
232     txtrect2 = pygame.rect.Rect((0, 400), (800, 50))
233     pygame.draw.rect(background, (255, 255, 255), txtrect2)
234     background.blit(txt2, txtrect2)
235     pygame.display.update()
236     pygame.time.wait(500)
237     end(manager)
238 def loser(manager):
239     print("Computer is the WINNER!!!")
240     f = pygame.font.Font('freesansbold.ttf', 100)
241     txt = f.render(str('LOST'), 1, (255, 255, 255), '#000000')
242     txtrect = pygame.rect.Rect((0, 300), (800, 80))
243     pygame.draw.rect(background, (0,0,0), txtrect)
244     background.blit(txt, txtrect)
245     pygame.display.update()
246     end(manager)
247 def tie(manager):
248     print("Tie !!!")
249     f = pygame.font.Font('freesansbold.ttf', 100)
250     txt = f.render(str('TIE'), 1, (255, 255, 255), '#4b2d05')
251     txtrect = pygame.rect.Rect((0, 300), (800, 80))
252     pygame.draw.rect(background, (75,45,5), txtrect)
253     background.blit(txt, txtrect)
254     pygame.display.update()
255     end(manager)
256 is_running = True
257
258 while is_running:
259     time_delta = clock.tick(60) / 1000.0
260     pygame.time.delay(100)
261     for event in pygame.event.get():
262
263         if event.type == pygame.QUIT:
264             is_running = False
265
266         if event.type == pygame.USEREVENT:
267             if event.user_type == 'ui_button_pressed':
268                 if event.ui_element == Start_button:
269                     manager.clear_and_reset()
270                     leave_button = pygame_gui.elements.UIButton(relative_rect=pygame.Rect((700, 550), (85, 35)),text='Leave', manager=manager)
271                     play_button = pygame_gui.elements.UIButton(relative_rect=pygame.Rect((700, 275), (90, 40)),text='Play', manager=manager)
272                     option_3_button = pygame_gui.elements.UIButton(relative_rect=pygame.Rect((225, 300), (50, 40)),text='3', manager=manager)
273                     option_5_button = pygame_gui.elements.UIButton(relative_rect=pygame.Rect((275, 300), (50, 40)),text='5', manager=manager)
274                     option_10_button = pygame_gui.elements.UIButton(relative_rect=pygame.Rect((325, 300), (50, 40)),text='10', manager=manager)
275                     option_full_game_button = pygame_gui.elements.UIButton(relative_rect=pygame.Rect((375, 300), (100, 40)),text='Full Game', manager=manager)
276                     manager.update(time_delta)
277                     decks = distributing(full_deck)
278                     game(manager,decks)
279                 if event.ui_element == play_button:
280                     draw_allowed = True
281                     shuffle_allowed = True

```

```

282     manager.clear_and_reset()
283     arena = pygame.Rect((142.5, 200), (500, 200))
284     pygame.draw.rect(background, (75, 45, 5), arena)
285     manager.update(time_delta)
286     leave_button = pygame_gui.elements.UIButton(relative_rect=pygame.Rect((700, 550), (85, 35)),text='Leave', manager=manager)
287     draw_button = pygame_gui.elements.UIButton(relative_rect=pygame.Rect((475, 475), (90, 40)),text='Draw', manager=manager)
288     shuffle_button = pygame_gui.elements.UIButton(relative_rect=pygame.Rect((205, 475), (90, 40)),text='Shuffle', manager=manager)
289     end_turn_button = pygame_gui.elements.UIButton(relative_rect=pygame.Rect((600, 475), (90, 40)),text='End Turn', manager=manager)
290     manager.update(time_delta)
291     c2 = computer_draw(deck2)
292     see_first_card(deck1)
293     if event.ui_element == Credit_button:
294         manager.clear_and_reset()
295         leave_button = pygame_gui.elements.UIButton(relative_rect=pygame.Rect((700, 550), (85, 35)),text='Leave', manager=manager)
296         f = pygame.font.Font('freesansbold.ttf', 25)
297         txt = f.render('Created by ELMOUHAJIR Yassir',1,(150, 0, 0),'#ffb72e')
298         txtrect = pygame.Rect.Rect((225, 295), (85, 35))
299         background.blit(txt, txtrect)
300         g = pygame.font.Font('freesansbold.ttf', 20)
301         txt0 = g.render('Supported by GIID, ENSAK', 1, (15, 225, 15), '#ffb72e')
302         txtrect0 = pygame.Rect.Rect((50, 500), (85, 35))
303         background.blit(txt0, txtrect0)
304     if event.ui_element == option_3_button:
305         limit = 3
306         print('First to 3 wins.')
307     if event.ui_element == option_5_button:
308         limit = 5
309         print('First to 5 wins.')
310     if event.ui_element == option_10_button:
311         limit = 10
312         print('First to 10 wins.')
313     if event.ui_element == option_full_game_button:
314         limit = 26
315         print("Player with the most cards at the end wins.")
316     if event.ui_element == leave_button or event.ui_element == menu_button:
317         manager.clear_and_reset()
318         pygame.display.update()
319         CARDS_WON = 0
320         CARDS_LOST = 0
321         background = pygame.Surface((800, 600))
322         background.fill(pygame.Color('#ffb72e'))
323         manager.update(time_delta)
324         manager.clear_and_reset()
325         Start_button = pygame_gui.elements.UIButton(relative_rect=pygame.Rect((x, y - 75), (100, 50)),text='START', manager=manager)
326         Credit_button = pygame_gui.elements.UIButton(relative_rect=pygame.Rect((x, y), (100, 50)),text='Credits', manager=manager)
327         Quit_button = pygame_gui.elements.UIButton(relative_rect=pygame.Rect((x, y + 75), (100, 50)),text='Quit', manager=manager)
328         decks = distributing(full_deck)
329     if event.ui_element == shuffle_button:
330         if shuffle_allowed == True:
331             deck1 = player1.shuffle(deck1)
332             deck_surface = pygame.Rect((322.5, 420), (126, 175))
333             pygame.draw.rect(background, pygame.Color('#ffb72e'), deck_surface)
334             e = 0
335             Player1_deck_surface = pygame.Rect((322.5, 420), (120, 168))
336             deck_img = pygame.image.load("JPEG\Green_back.jpg")
337             for i in deck1:
338                 background.blit(deck_img, Player1_deck_surface)
339                 pygame.time.delay(10)
340                 Player1_deck_surface = pygame.Rect((322.5 - e, 420 - e), (120, 168))
341                 e = e + 0.2
342             see_first_card(deck1)
343             shuffle_allowed = False
344             manager.update(time_delta)
345     if event.ui_element == draw_button:
346         if draw_allowed == True:
347             c1 = player1.draw_card()
348             adresse = "JPEG/" + c1[2]
349             cpath = pygame.image.load(adresse)
350             drawn_card_surface = pygame.Rect((220, 210), (120, 168))
351             deck_surface = pygame.Rect((322.5, 420), (126, 175))
352             pygame.draw.rect(background,pygame.Color('#ffb72e'),deck_surface)

```

```

353         e = 0
354         Player1_deck_surface = pygame.Rect((322.5, 420), (120, 168))
355         for i in deck1:
356             background.blit(deck_img, Player1_deck_surface)
357             Player1_deck_surface = pygame.Rect((322.5 + e, 420 + e), (120, 168))
358             e = e + 0.2
359         pygame.time.delay(50)
360         background.blit(cpath, drawn_card_surface)
361         manager.update(time_delta)
362         draw_allowed = False
363         shuffle_allowed = False
364         end_turn_allowed = True
365         manager.update(time_delta)
366     if event.ui_element == end_turn_button:
367         if end_turn_allowed == True:
368             pygame.time.wait(100)
369             if battle(c1, c2) == 1:
370                 won(E, deck_img)
371                 E = E + 0.2
372                 won(E, deck_img)
373                 E = E + 0.2
374                 CARDS_WON = CARDS_WON + 1
375                 player1_score(CARDS_WON)
376                 print("Player Wins !!")
377             else:
378                 lost(E, deck_img)
379                 E = E + 0.2
380                 lost(E, deck_img)
381                 E = E + 0.2
382                 CARDS_LOST = CARDS_LOST + 1
383                 computer_score(CARDS_LOST)
384                 print("Computer Wins !!")
385             see_first_card(deck1)
386             draw_allowed = True
387             shuffle_allowed = True
388             clear_arena()
389             c2 = computer_draw(deck2)
390             end_turn_allowed = False
391             manager.update(time_delta)
392     if event.ui_element == Quit_button:
393         exit()
394         pygame.display.quit()
395     if CARDS_WON >= limit:
396         winner(manager)
397         menu_button = pygame_gui.elements.UIButton(relative_rect=pygame.Rect((675, 535), (90, 45)),text='Main Menu', manager=manager)
398         CARDS_WON = 0
399         CARDS_LOST = 0
400     if CARDS_LOST >= limit:
401         loser(manager)
402         menu_button = pygame_gui.elements.UIButton(relative_rect=pygame.Rect((675, 535), (90, 45)),text='Main Menu', manager=manager)
403         CARDS_LOST = 0
404         CARDS_WON = 0
405     if deck1 == [] and CARDS_WON > CARDS_LOST:
406         winner(manager)
407         menu_button = pygame_gui.elements.UIButton(relative_rect=pygame.Rect((675, 535), (90, 45)),text='Main Menu', manager=manager)
408         CARDS_WON = 0
409         CARDS_LOST = 0
410     if (deck2 == [] or deck1 == []) and CARDS_WON < CARDS_LOST:
411         loser(manager)
412         menu_button = pygame_gui.elements.UIButton(relative_rect=pygame.Rect((675, 535), (90, 45)),text='Main Menu', manager=manager)
413         CARDS_LOST = 0
414         CARDS_WON = 0
415         player1 = player_human('Player1', deck1)
416     if (deck2 == [] or deck1 == []) and CARDS_WON < CARDS_LOST:
417         tie(manager)
418         menu_button = pygame_gui.elements.UIButton(relative_rect=pygame.Rect((675, 535), (90, 45)),text='Main Menu', manager=manager)
419         CARDS_LOST = 0
420         CARDS_WON = 0
421         player1 = player_human('Player1',deck1)
422
423     manager.process_events(event)

```

```
424     manager.update(time_delta)
425     window_surface.blit(background, (0, 0))
426     manager.draw_ui(window_surface)
427     pygame.display.update()
428     pygame.quit()
429
430
431
```