

Projet du groupe 1 en Outils informatiques collaboratifs

William Nizet - Kévin Martins Da Veiga - Felipe Pereira Manfrin - Anri Kennel

Table des matières

1	Cahier des charges	2
2	Spécifications fonctionnelles	2
3	Plan de réalisation	2
4	Code	2

1 Cahier des charges

- Le jeu va s'inspirer du Blackjack
- Il y a un croupier, un joueur et une pioche
- Une version où la mise est remplacée par des points
- On commence avec 10 points, -2 si le joueur perd, +2 si le joueur gagne
- Sans interface graphique

2 Spécifications fonctionnelles

- Language de programmation : C
- Avec la possibilité de tirer une carte, rester ou doubler la mise avec un maximum de 5 cartes
- On a une main de 4 cartes au début de la partie
- Une pioche générée aléatoirement
- Un croupier qui pioche tant qu'il a moins de 16 points
- Deux niveaux, un croupier qui veut tout le temps arriver à 21 points (le max), et un croupier qui veut jouer moins dangereusement
- Fin de jeu : Soit proche de 21, soit 21 pile, soit plus de 21. S'il dépasse, il perd. Le plus proche de 21 gagne.

3 Plan de réalisation

- La personne A s'occupe du tirage des cartes
- La personne B s'occupe du croupier
- La personne C s'occupe du joueur
- La personne D s'occupe du système de points

4 Code

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <conio.h>
4
5 int cards[53] = {1,2,3,4,5,6,7,8,9,10,11,12,13,
6                 14,15,16,17,18,19,20,21,22,23,24,25,26,
7                 27,28,29,30,31,32,33,34,35,36,37,38,39,
8                 40,41,42,43,44,45,46,47,48,49,50,51,52};
9
10 int player1[10];
11 int player2[10];
12 int dealer[10];
13
14 int real_value(int nb) {
```

```

15 |     if (nb == 1 || nb == 14 || nb == 27 || nb == 40) {
16 |         return 1;
17 |     } else if (nb == 2 || nb == 15 || nb == 28 || nb == 41) {
18 |         return 2;
19 |     } else if (nb == 3 || nb == 16 || nb == 29 || nb == 42) {
20 |         return 3;
21 |     } else if (nb == 4 || nb == 17 || nb == 30 || nb == 43) {
22 |         return 4;
23 |     } else if (nb == 5 || nb == 18 || nb == 31 || nb == 44) {
24 |         return 5;
25 |     } else if (nb == 6 || nb == 19 || nb == 32 || nb == 45) {
26 |         return 6;
27 |     } else if (nb == 7 || nb == 20 || nb == 33 || nb == 46) {
28 |         return 7;
29 |     } else if (nb == 8 || nb == 21 || nb == 34 || nb == 47) {
30 |         return 8;
31 |     } else if (nb == 9 || nb == 22 || nb == 35 || nb == 48) {
32 |         return 9;
33 |     } else if (nb == 10 || nb == 23 || nb == 36 || nb == 49) {
34 |         return 10;
35 |     } else if (nb == 11 || nb == 24 || nb == 37 || nb == 50) {
36 |         return 10;
37 |     } else if (nb == 12 || nb == 25 || nb == 38 || nb == 51) {
38 |         return 10;
39 |     } else if (nb == 13 || nb == 26 || nb == 39 || nb == 52) {
40 |         return 10;
41 |     }
42 | }
43
44 | char real_color(int nb) {
45 |     if (nb >= 1 && nb <= 13) {
46 |         return '*';
47 |     } else if (nb >= 14 && nb <= 26) {
48 |         return '#';
49 |     } else if (nb >= 27 && nb <= 39) {
50 |         return '&';
51 |     } else {
52 |         return '$';
53 |     }
54 | }
55
56 | int victory(int sum) {
57 |     if (sum <= 21) {
58 |         return 0;
59 |     } else if (sum > 21) {
60 |         return 1;

```

```

61     }
62 }
63
64 void draw_card(int value, char couleur) {
65     if (value == 1 || value == 14 || value == 27 || value == 40) {
66         printf("-----\n");
67         printf("| A %c |\n",couleur);
68         printf("| |\n");
69         printf("| |\n");
70         printf("| |\n");
71         printf("| |\n");
72         printf("| |\n");
73         printf("| A %c|\n",couleur);
74         printf("-----\n\n");
75     } else if (value == 11 || value == 24 || value == 37 || value == 50) {
76         printf("-----\n");
77         printf("| J %c |\n",couleur);
78         printf("| |\n");
79         printf("| |\n");
80         printf("| |\n");
81         printf("| |\n");
82         printf("| |\n");
83         printf("| J %c|\n",couleur);
84         printf("-----\n\n");
85     } else if (value == 12 || value == 25 || value == 38 || value == 51) {
86         printf("-----\n");
87         printf("| Q %c |\n",couleur);
88         printf("| |\n");
89         printf("| |\n");
90         printf("| |\n");
91         printf("| |\n");
92         printf("| |\n");
93         printf("| Q %c|\n",couleur);
94         printf("-----\n\n");
95     } else if (value == 13 || value == 26 || value == 39 || value == 52) {
96         printf("-----\n");
97         printf("| K %c |\n",couleur);
98         printf("| |\n");
99         printf("| |\n");
100        printf("| |\n");
101        printf("| |\n");
102        printf("| |\n");
103        printf("| K %c|\n",couleur);
104        printf("-----\n\n");
105    } else {
106        printf("-----\n");

```

```

107     printf("|%2d %c |\n",real_value(value),couleur);
108     printf("|\n");
109     printf("|\n");
110     printf("|\n");
111     printf("|\n");
112     printf("|\n");
113     printf("| %2d %c|\n");
114     printf("-----\n\n");
115     }
116
117 }
118
119 int main() {
120     int i = 0, card_acc, acc = 2, init_dsum;
121     int res_p1, sum1 = 0;
122     int res_p2, sum2 = 0;
123
124     printf("Welcome to BLACKJACK !\n");
125     printf("Press any key to continue...\n\n");
126     getch();
127
128     printf(" -> Your goal is to score exactly 21 points or the nearest possible
           to 21.\n");
129     sleep(1);
130     printf(" -> You cannot exceed 21 points.\n");
131     sleep(1);
132     printf(" -> You must have a final score greater than the dealer's one.\n"
           );
133     sleep(1);
134     printf(" -> If the dealer's final score is greater than 21, everyone who has
           a score less or equal to 21 wins the game.\n\n");
135     sleep(1);
136     printf("Are you ready ?\n\n");
137     printf("Press any key to continue...\n");
138     getch();
139     system("cls");
140
141     printf("The game begins in...\n");
142     printf("3\n");
143     sleep(1);
144     printf("2\n");
145     sleep(1);
146     printf("1\n");
147     sleep(1);
148     printf("GO!");
149     sleep(1);

```

```

150     system("cls");
151
152     srand(time(NULL));
153     dealer[0] = (rand() % 52) + 1;
154
155     dealer[1] = (rand() % 52) + 1;
156
157     cards[dealer[0]-1] = 0;
158     cards[dealer[1]-1] = 0;
159
160     printf("Dealer first card is : \n\n");
161     sleep(1);
162     draw_card(dealer[0],real_color(dealer[0]));
163     sleep(1);
164     printf("Dealer second card is : ");
165     sleep(1);
166     printf("hidden card\n\n");
167     sleep(1);
168     printf("Press any key to continue...\n");
169     getch();
170     system("cls");
171
172     printf("Each player will now receive their two first initial cards :\n\n");
173     sleep(2);
174
175
176     player1[0] = (rand() % 52) + 1;
177     if (cards[player1[0]-1] == 0) {
178         while (cards[player1[0]-1] == 0) {
179             player1[0] = (rand() % 52) + 1;
180         }
181         cards[player1[0]-1] = 0;
182     } else {
183         cards[player1[0]-1] = 0;
184     }
185
186
187     player1[1] = (rand() % 52) + 1;
188     if (cards[player1[1]-1] == 0) {
189         while (cards[player1[1]-1] == 0) {
190             player1[1] = (rand() % 52) + 1;
191         }
192         cards[player1[1]-1] = 0;
193     } else {
194         cards[player1[1]-1] = 0;
195     }

```

```

196
197
198 player2[0] = (rand() % 52) + 1;
199 if (cards[player2[0]-1] == 0) {
200     while (cards[player2[0]-1] == 0) {
201         player2[0] = (rand() % 52) + 1;
202     }
203     cards[player2[0]-1] = 0;
204 } else {
205     cards[player2[0]-1] = 0;
206 }
207
208
209 player2[1] = (rand() % 52) + 1;
210 if (cards[player2[1]-1] == 0) {
211     while (cards[player2[1]-1] == 0) {
212         player2[1] = (rand() % 52) + 1;
213     }
214     cards[player2[1]-1] = 0;
215 } else {
216     cards[player2[1]-1] = 0;
217 }
218
219 printf("Player 1 first cards are : \n\n");
220 sleep(1);
221 draw_card(player1[0],real_color(player1[0]));
222 sleep(1);
223 draw_card(player1[1],real_color(player1[1]));
224 sleep(1);
225 printf("Player 2 first cards are : \n\n");
226 sleep(1);
227 draw_card(player2[0],real_color(player2[0]));
228 sleep(1);
229 draw_card(player2[1],real_color(player2[1]));
230
231 printf("Press any key to continue...");
232 getch();
233 system("cls");
234
235 printf("Player 1, you can now choose either to turn more cards up or to
      keep your current values : \n\n");
236 sleep(2);
237
238 printf("Your current cards are : \n\n");
239 draw_card(player1[0],real_color(player1[0]));
240 draw_card(player1[1],real_color(player1[1]));

```

```

241 | printf("Would you like to turn up more cards ? Enter '1' for YES or '2'
      |     for NO : \n");
242 | scanf("%d", &res_p1);
243 | if (res_p1 == 1) {
244 |     while(res_p1 == 1) {
245 |         card_acc = (rand() % 52) + 1;
246 |         if (cards[card_acc-1] == 0) {
247 |             while(cards[card_acc-1] == 0) {
248 |                 card_acc = (rand() % 52) + 1;
249 |             }
250 |             cards[card_acc-1] = 0;
251 |         } else {
252 |             cards[card_acc-1] = 0;
253 |         }
254 |         player1[acc] = card_acc;
255 |         ++acc;
256 |
257 |         while (player1[i] != 0) {
258 |             sum1 += real_value(player1[i]);
259 |             ++i;
260 |         }
261 |
262 |         printf("Your next card is : \n\n");
263 |         sleep(1);
264 |         draw_card(card_acc,real_color(card_acc));
265 |         sleep(1);
266 |
267 |         if (victory(sum1) == 1) {
268 |             printf(" -> Your final cards are : <-\n\n");
269 |             for (int a = 0; player1[a] != 0; ++a) {
270 |                 draw_card(player1[a],real_color(player1[a]));
271 |             }
272 |             printf("\n");
273 |             printf("Your final score is %d, which is greater than 21. You've
      |                 lost the game !\n",sum1);
274 |             sum1 = 0;
275 |             break;
276 |         } else {
277 |             printf("Would you like to turn up more cards ? Enter '1' for
      |                 YES or '2' for NO : \n");
278 |             scanf("%d", &res_p1);
279 |         }
280 |     }
281 |     if (sum1 != 0) {
282 |         printf("\n");

```



```

283         printf("You decided to stop turning up new cards. Your final
           score is : ");
284         sleep(1);
285         printf("%d\n",sum1);
286     }
287
288 } else {
289     while (player1[i] != 0) {
290         sum1 += real_value(player1[i]);
291         ++i;
292     }
293     printf("\n");
294     printf("You decided to keep your current values. You can no longer
           have more cards. Your final score is : %d\n", sum1);
295 }
296 printf("Press any key to continue...");
297 getch();
298 system("cls");
299
300 i = 0;
301 acc = 2;
302
303 printf("Player 2, you can now choose either to turn more cards up or to
           keep your current values :\n\n");
304 sleep(1);
305
306 printf("Your current cards are : \n\n");
307 draw_card(player2[0],real_color(player2[0]));
308 draw_card(player2[1],real_color(player2[1]));
309 printf("Would you like to turn up more cards ? Enter '1' for YES or '2'
           for NO : \n");
310 scanf("%d", &res_p2);
311 if (res_p2 == 1) {
312     while(res_p2 == 1) {
313         card_acc = (rand() % 52) + 1;
314         if (cards[card_acc-1] == 0) {
315             while(cards[card_acc-1] == 0) {
316                 card_acc = (rand() % 52) + 1;
317             }
318             cards[card_acc-1] = 0;
319         } else {
320             cards[card_acc-1] = 0;
321         }
322         player2[acc] = card_acc;
323         ++acc;
324

```

```

325     while (player2[i] != 0) {
326         sum2 += real_value(player2[i]);
327         ++i;
328     }
329
330     printf("Your next card is : \n\n");
331     sleep(1);
332     draw_card(card_acc,real_color(card_acc));
333     sleep(1);
334
335     if (victory(sum2) == 1) {
336         printf(" -> Your final cards are : <-\n\n");
337         for (int b = 0; player2[b] != 0; ++b) {
338             draw_card(player2[b],real_color(player2[b]));
339         }
340         printf("\n");
341         printf("Your final score is %d, which is greater than 21. You've
342             lost the game !\n",sum2);
343         sum2 = 0;
344         break;
345     } else {
346         printf("Would you like to turn up more cards ? Enter '1' for
347             YES or '2' for NO : \n");
348         scanf("%d", &res_p2);
349     }
350     if (sum2 != 0) {
351         printf("\n");
352         printf("You decided to stop turning up new cards. Your final
353             score is : ");
354         sleep(1);
355         printf("%d\n",sum2);
356     }
357     } else {
358         while (player2[i] != 0) {
359             sum2 += real_value(player2[i]);
360             ++i;
361         }
362         printf("\n");
363         printf("You decided to keep your current values. You can no longer
364             have more cards. Your final score is : %d\n", sum2);
365     }
366     printf("Press any key to continue...");
367     getch();
368     system("cls");

```

```

367
368     acc = 2;
369
370     printf("--- THIS IS THE DEALER'S TURN ---\n\n");
371     sleep(2);
372
373     init_dsum = real_value(dealer[0]) + real_value(dealer[1]);
374     printf("Dealer current cards are : \n\n");
375     sleep(1);
376     draw_card(dealer[0],real_color(dealer[0]));
377     sleep(1);
378     draw_card(dealer[1],real_color(dealer[1]));
379     sleep(2);
380
381     if (init_dsum < 16) {
382         printf("Dealer's score is less than 16. He will now turn up as much as
383             necessary cards until he gets a score greater than 16.\n\n");
384         sleep(2);
385         while (init_dsum < 16) {
386             init_dsum = 0;
387             card_acc = (rand() % 52) + 1;
388             if (cards[card_acc-1] == 0) {
389                 while(cards[card_acc-1] == 0) {
390                     card_acc = (rand() % 52) + 1;
391                 }
392                 cards[card_acc-1] = 0;
393             } else {
394                 cards[card_acc-1] = 0;
395             }
396             dealer[acc] = card_acc;
397             ++acc;
398             for (int c = 0; dealer[c] != 0; ++c) {
399                 init_dsum += real_value(dealer[c]);
400             }
401             printf("Dealer's hand is : \n\n");
402             draw_card(dealer[0],real_color(dealer[0]));
403             draw_card(dealer[1],real_color(dealer[1]));
404             sleep(1);
405             for (int d = 2; dealer[d] != 0; ++d) {
406                 draw_card(dealer[d],real_color(dealer[d]));
407                 sleep(1);
408             }
409             printf("\n");
410             sleep(1);
411             printf("Dealer's final score is : %d\n\n",init_dsum);

```

```

412     sleep(1);
413
414     if (init_dsum > 21) {
415         printf("NO WAY ! Dealer's score is greater than 21 ! Everyone who
         has a final score less than 21 wins the game !\n\n");
416         sleep(1);
417         if (sum1 != 0) {
418             printf("Player 1 wins the game! --- BLACKJACK ---\n");
419         } else {
420             printf("Player 1 loses the game!\n\n");
421         }
422         if (sum2 != 0) {
423             printf("Player 2 wins the game! --- BLACKJACK ---\n");
424         } else {
425             printf("Player 2 loses the game!\n\n");
426         }
427     } else {
428         printf("You win the game if you have a final score greater than %d
         .\n\n",init_dsum);
429         sleep(2);
430         if (sum1 != 0 && sum1 > init_dsum) {
431             printf("Player 1 wins the game with %d points! ->
         BLACKJACK <-\n",sum1);
432         } else {
433             printf("Player 1 final score is less than dealer's score!\n");
434             printf("Player 1 loses the game!\n\n");
435         }
436         if (sum2 != 0 && sum2 > init_dsum) {
437             printf("Player 2 wins the game with %d points! ->
         BLACKJACK <-\n",sum2);
438         } else {
439             printf("Player 2 final score is less than dealer's score!\n");
440             printf("Player 2 loses the game!\n\n");
441         }
442     }
443 } else {
444     printf("You win the game if you have a final score greater than %d.\n
         \n",init_dsum);
445     sleep(2);
446     if (sum1 != 0 && sum1 > init_dsum) {
447         printf("Player 1 wins the game with %d points! ->
         BLACKJACK <-\n",sum1);
448     } else if (sum1 > 0) {
449         printf("Player 1 final score is less than dealer's score!\n");
450         printf("Player 1 loses the game!\n\n");
451     } else {

```

```

452         printf("Player 1 final score is greater than 21!\n");
453         printf("Player 1 loses the game!\n\n");
454     }
455     if (sum2 != 0 && sum2 > init_dsum) {
456         printf("Player 2 wins the game with %d points! ->
         BLACKJACK <-\n\n",sum2);
457     } else if (sum2 > 0) {
458         printf("Player 2 final score is less than dealer's score!\n");
459         printf("Player 2 loses the game!\n\n");
460     } else {
461         printf("Player 2 final score is greater than 21!\n");
462         printf("Player 2 loses the game!\n\n");
463     }
464 }
465
466 printf("Press any key to continue...");
467 getch();
468
469 return 0;
470 }

```