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

— 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
4  // linux
5  #include <curses.h> // fonction de conio.h en linux
6  #include <unistd.h> // fonction sleep
7  #include <time.h> // fonction time
8
9  // windows
10 // #include <conio.h>
11
12 int cards[53] = {1,2,3,4,5,6,7,8,9,10,11,12,13,
13                  14,15,16,17,18,19,20,21,22,23,24,25,26,
14                  27,28,29,30,31,32,33,34,35,36,37,38,39,
```

```
15                  40,41,42,43,44,45,46,47,48,49,50,51,52};
16
17  int player1[10];
18  int player2[10];
19  int dealer[10];
20
21  int real_value(int nb) {
22      if (nb == 1 || nb == 14 || nb == 27 || nb == 40) {
23          return 1;
24      } else if (nb == 2 || nb == 15 || nb == 28 || nb == 41) {
25          return 2;
26      } else if (nb == 3 || nb == 16 || nb == 29 || nb == 42) {
27          return 3;
28      } else if (nb == 4 || nb == 17 || nb == 30 || nb == 43) {
29          return 4;
30      } else if (nb == 5 || nb == 18 || nb == 31 || nb == 44) {
31          return 5;
32      } else if (nb == 6 || nb == 19 || nb == 32 || nb == 45) {
33          return 6;
34      } else if (nb == 7 || nb == 20 || nb == 33 || nb == 46) {
35          return 7;
36      } else if (nb == 8 || nb == 21 || nb == 34 || nb == 47) {
37          return 8;
38      } else if (nb == 9 || nb == 22 || nb == 35 || nb == 48) {
39          return 9;
40      } else if (nb == 10 || nb == 23 || nb == 36 || nb == 49) {
41          return 10;
42      } else if (nb == 11 || nb == 24 || nb == 37 || nb == 50) {
43          return 10;
44      } else if (nb == 12 || nb == 25 || nb == 38 || nb == 51) {
45          return 10;
46      } else if (nb == 13 || nb == 26 || nb == 39 || nb == 52) {
47          return 10;
48      }
49
50      return 0;
51  }
52
53  char real_color(int nb) {
54      if (nb >= 1 && nb <= 13) {
55          return '*';
56      } else if (nb >= 14 && nb <= 26) {
57          return '#';
58      } else if (nb >= 27 && nb <= 39) {
59          return '&';
60      } else {
```

```
61          return '$';
62      }
63  }
64
65  int victory(int sum) {
66      if (sum <= 21) {
67          return 0;
68      } else if (sum > 21) {
69          return 1;
70      }
71
72      return 2;
73  }
74
75  void draw_card(int value, char couleur) {
76      if (value == 1 || value == 14 || value == 27 || value == 40) {
77          printf(" ------------\n");
78      printf("|A %c |\n",couleur);
79      printf("| |\n");
80      printf("| |\n");
81      printf("| |\n");
82      printf("| |\n");
83      printf("| |\n");
84      printf("| |\n");
85      printf("| A %c|\n",couleur);
86      printf(" ------------\n\n");
87      } else if (value == 11 || value == 24 || value == 37 || value == 50) {
88          printf(" ------------\n");
89      printf("|J %c |\n",couleur);
90      printf("| |\n");
91      printf("| |\n");
92      printf("| |\n");
93      printf("| |\n");
94      printf("| |\n");
95      printf("| |\n");
96      printf("| J %c|\n",couleur);
97      printf(" ------------\n\n");
98      } else if (value == 12 || value == 25 || value == 38 || value == 51) {
99          printf(" ------------\n");
100     printf("|Q %c |\n",couleur);
101     printf("| |\n");
102     printf("| |\n");
103     printf("| |\n");
104     printf("| |\n");
105     printf("| |\n");
106     printf("| |\n");
```

```
107         printf("| Q %c|\n",couleur);
108         printf(" ————————————\n\n");
109         } else if (value == 13 || value == 26 || value == 39 || value == 52) {
110             printf(" ————————————\n");
111         printf("|K %c |\n",couleur);
112         printf("| |\n");
113         printf("| |\n");
114         printf("| |\n");
115         printf("| |\n");
116         printf("| |\n");
117         printf("| |\n");
118         printf("| K %c|\n",couleur);
119         printf(" ————————————\n\n");
120         } else {
121             printf(" ————————————\n");
122         printf("|%2d %c |\n",real_value(value),couleur);
123         printf("| |\n");
124         printf("| |\n");
125         printf("| |\n");
126         printf("| |\n");
127         printf("| |\n");
128         printf("| |\n");
129         printf("| %2d %c|\n",real_value(value),couleur);
130         printf(" ————————————\n\n");
131         }
132 
133 }
134 
135 int main() {
136     int i = 0, card_acc, acc = 2, init_dsum;
137     int res_p1, sum1 = 0;
138     int res_p2, sum2 = 0;
139 
140     printf("Welcome to BLACKJACK !\n");
141     // linux
142     printf("Press ENTER to continue...\n\n");
143     getchar();
144 
145     // windows
146     // printf("Press any key to continue...\n\n");
147     // getch();
148 
149     printf(" -> Your goal is to score exactly 21 points or the nearest possible
            to 21.\n");
150     sleep(1);
151     printf(" -> You cannot exceed 21 points.\n");
```

```
152     sleep(1);
153     printf(" -> You must have a final score greater than the dealer's one.\n"
            );
154     sleep(1);
155     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");
156     sleep(1);
157     printf("Are you ready ?\n\n");
158     // linux
159     printf("Press ENTER to continue...\n\n");
160     getchar();
161     system("clear");
162
163     // windows
164     // printf("Press any key to continue...\n\n");
165     // getch();
166     //system("cls");
167
168     printf("The game begins in...\n");
169     printf("3\n");
170     sleep(1);
171     printf("2\n");
172     sleep(1);
173     printf("1\n");
174     sleep(1);
175     printf("GO!\n");
176     sleep(1);
177     // linux
178     system("clear");
179
180     // windows
181     // system("cls");
182
183     srand(time(NULL));
184     dealer[0] = (rand() % 52) + 1;
185
186     dealer[1] = (rand() % 52) + 1;
187
188     cards[dealer[0]-1] = 0;
189     cards[dealer[1]-1] = 0;
190
191     printf("Dealer first card is : \n\n");
192     sleep(1);
193     draw_card(dealer[0],real_color(dealer[0]));
194     sleep(1);
195     printf("Dealer second card is : ");
```

```
196     sleep(1);
197     printf("hidden card\n\n");
198     sleep(1);
199     // linux
200     printf("Press ENTER to continue...\n\n");
201     getchar();
202     system("clear");
203
204     // windows
205     // printf("Press any key to continue...\n\n");
206     // getch();
207     // system("cls");
208
209     printf("Each player will now receive their two first initial cards :\n\n");
210     sleep(2);
211
212
213     player1[0] = (rand() % 52) + 1;
214     if (cards[player1[0]-1] == 0) {
215         while (cards[player1[0]-1] == 0) {
216             player1[0] = (rand() % 52) + 1;
217         }
218         cards[player1[0]-1] = 0;
219     } else {
220         cards[player1[0]-1] = 0;
221     }
222
223
224     player1[1] = (rand() % 52) + 1;
225     if (cards[player1[1]-1] == 0) {
226         while (cards[player1[1]-1] == 0) {
227             player1[1] = (rand() % 52) + 1;
228         }
229         cards[player1[1]-1] = 0;
230     } else {
231         cards[player1[1]-1] = 0;
232     }
233
234
235     player2[0] = (rand() % 52) + 1;
236     if (cards[player2[0]-1] == 0) {
237         while (cards[player2[0]-1] == 0) {
238             player2[0] = (rand() % 52) + 1;
239         }
240         cards[player2[0]-1] = 0;
241     } else {
```

```
242             cards[player2[0]-1] = 0;
243         }
244
245
246         player2[1] = (rand() % 52) + 1;
247         if (cards[player2[1]-1] == 0) {
248             while (cards[player2[1]-1] == 0) {
249                 player2[1] = (rand() % 52) + 1;
250             }
251             cards[player2[1]-1] = 0;
252         } else {
253             cards[player2[1]-1] = 0;
254         }
255
256         printf("Player 1 first cards are : \n\n");
257         sleep(1);
258         draw_card(player1[0],real_color(player1[0]));
259         sleep(1);
260         draw_card(player1[1],real_color(player1[1]));
261         sleep(1);
262         printf("Player 2 first cards are : \n\n");
263         sleep(1);
264         draw_card(player2[0],real_color(player2[0]));
265         sleep(1);
266         draw_card(player2[1],real_color(player2[1]));
267
268         // linux
269         printf("Press ENTER to continue...\n\n");
270         getchar();
271         system("clear");
272
273         // windows
274         // printf("Press any key to continue...\n\n");
275         // getch();
276         // system("cls");
277
278         printf("Player 1, you can now choose either to turn more cards up or to
                keep your current values :\n\n");
279         sleep(2);
280
281         printf("Your current cards are : \n\n");
282         draw_card(player1[0],real_color(player1[0]));
283         draw_card(player1[1],real_color(player1[1]));
284         printf("Would you like to turn up more cards ? Enter '1' for YES or '2'
                for NO : \n");
285         scanf("%d", &res_p1);
```

```
286     if (res_p1 == 1) {
287         while(res_p1 == 1) {
288             card_acc = (rand() % 52) + 1;
289             if (cards[card_acc-1] == 0) {
290                 while(cards[card_acc-1] == 0) {
291                     card_acc = (rand() % 52) + 1;
292                 }
293                 cards[card_acc-1] = 0;
294             } else {
295                 cards[card_acc-1] = 0;
296             }
297             player1[acc] = card_acc;
298             ++acc;
299 
300             while (player1[i] != 0) {
301                 sum1 += real_value(player1[i]);
302                 ++i;
303             }
304 
305             printf("Your next card is : \n\n");
306             sleep(1);
307             draw_card(card_acc,real_color(card_acc));
308             sleep(1);
309 
310             if (victory(sum1) == 1) {
311                 printf(" -> Your final cards are : <-\n\n");
312                 for (int a = 0; player1[a] != 0; ++a) {
313                     draw_card(player1[a],real_color(player1[a]));
314                 }
315             printf("\n");
316             printf("Your final score is %d, which is greater than 21. You've
                     lost the game !\n",sum1);
317             sum1 = 0;
318             break;
319             } else {
320                 printf("Would you like to turn up more cards ? Enter '1' for
                         YES or '2' for NO : \n");
321                 scanf("%d", &res_p1);
322             }
323         }
324         if (sum1 != 0) {
325             printf("\n");
326             printf("You decided to stop turning up new cards. Your final
                     score is : ");
327             sleep(1);
328             printf("%d\n",sum1);
```

```
329             }
330             // linux
331             printf("Press ENTER to continue...\n\n");
332             getchar();
333             system("clear");
334
335             // windows
336             // printf("Press any key to continue...\n\n");
337             // getch();
338             // system("cls");
339
340         } else {
341             while (player1[i] != 0) {
342                     sum1 += real_value(player1[i]);
343                     ++i;
344                     }
345             printf("\n");
346             printf("You decided to keep your current values. You can no longer
                    have more cards. Your final score is : %d\n", sum1);
347         }
348         // linux
349         printf("Press ENTER to continue...\n\n");
350         getchar();
351         system("clear");
352
353         // windows
354         // printf("Press any key to continue...\n\n");
355         // getch();
356         // system("cls");
357
358         i = 0;
359         acc = 2;
360
361         printf("Player 2, you can now choose either to turn more cards up or to
                    keep your current values :\n\n");
362         sleep(1);
363
364         printf("Your current cards are : \n\n");
365         draw_card(player2[0],real_color(player2[0]));
366         draw_card(player2[1],real_color(player2[1]));
367         printf("Would you like to turn up more cards ? Enter '1' for YES or '2'
                    for NO : \n");
368         scanf("%d", &res_p2);
369         if (res_p2 == 1) {
370             while(res_p2 == 1) {
371                 card_acc = (rand() % 52) + 1;
```

```
372             if (cards[card_acc-1] == 0) {
373                 while(cards[card_acc-1] == 0) {
374                     card_acc = (rand() % 52) + 1;
375                 }
376                 cards[card_acc-1] = 0;
377             } else {
378                 cards[card_acc-1] = 0;
379             }
380             player2[acc] = card_acc;
381             ++acc;
382
383             while (player2[i] != 0) {
384                 sum2 += real_value(player2[i]);
385                 ++i;
386             }
387
388             printf("Your next card is : \n\n");
389             sleep(1);
390             draw_card(card_acc,real_color(card_acc));
391             sleep(1);
392
393             if (victory(sum2) == 1) {
394                 printf(" -> Your final cards are : <-\n\n");
395                 for (int b = 0; player2[b] != 0; ++b) {
396                     draw_card(player2[b],real_color(player2[b]));
397                 }
398             printf("\n");
399             printf("Your final score is %d, which is greater than 21. You've
                     lost the game !\n",sum2);
400             sum2 = 0;
401             break;
402             } else {
403                 printf("Would you like to turn up more cards ? Enter '1' for
                         YES or '2' for NO : \n");
404                 scanf("%d", &res_p2);
405             }
406         }
407         if (sum2 != 0) {
408             printf("\n");
409             printf("You decided to stop turning up new cards. Your final
                     score is : ");
410             sleep(1);
411             printf("%d\n",sum2);
412         }
413
414     } else {
```

```
415             while (player2[i] != 0) {
416                     sum2 += real_value(player2[i]);
417                     ++i;
418                     }
419             printf("\n");
420             printf("You decided to keep your current values. You can no longer
                    have more cards. Your final score is : %d\n", sum2);
421     }
422     // linux
423     printf("Press ENTER to continue...\n\n");
424     getchar();
425     system("clear");
426
427     // windows
428     // printf("Press any key to continue...\n\n");
429     // getch();
430     // system("cls");
431
432     acc = 2;
433
434     printf("-- THIS IS THE DEALER'S TURN --\n\n");
435     sleep(2);
436
437     init_dsum = real_value(dealer[0]) + real_value(dealer[1]);
438     printf("Dealer current cards are : \n\n");
439     sleep(1);
440     draw_card(dealer[0],real_color(dealer[0]));
441     sleep(1);
442     draw_card(dealer[1],real_color(dealer[1]));
443     sleep(2);
444
445     if (init_dsum < 16) {
446             printf("Dealer's score is less than 16. He will now turn up as much as
                    necessary cards until he gets a score greater than 16.\n\n");
447             sleep(2);
448             while (init_dsum < 16) {
449             init_dsum = 0;
450             card_acc = (rand() % 52) + 1;
451             if (cards[card_acc-1] == 0) {
452                     while(cards[card_acc-1] == 0) {
453                             card_acc = (rand() % 52) + 1;
454                             }
455                     cards[card_acc-1] = 0;
456             } else {
457                     cards[card_acc-1] = 0;
458             }
```

```
459             dealer[acc] = card_acc;
460             ++acc;
461             for (int c = 0; dealer[c] != 0; ++c) {
462                 init_dsum += real_value(dealer[c]);
463             }
464         }
465         printf("Dealer's hand is : \n\n");
466         draw_card(dealer[0],real_color(dealer[0]));
467         draw_card(dealer[1],real_color(dealer[1]));
468         sleep(1);
469         for (int d = 2; dealer[d] != 0; ++d) {
470             draw_card(dealer[d],real_color(dealer[d]));
471             sleep(1);
472         }
473         printf("\n");
474         sleep(1);
475         printf("Dealer's final score is : %d\n\n",init_dsum);
476         sleep(1);
477 
478         if (init_dsum > 21) {
479             printf("NO WAY ! Dealer's score is greater than 21 ! Everyone who
                        has a final score less than 21 wins the game !\n\n");
480             sleep(1);
481             if (sum1 != 0) {
482                 printf("Player 1 wins the game! -- BLACKJACK --\n");
483             } else {
484                 printf("Player 1 loses the game!\n\n");
485             }
486             if (sum2 != 0) {
487                 printf("Player 2 wins the game! -- BLACKJACK --\n");
488             } else {
489                 printf("Player 2 loses the game!\n\n");
490             }
491         } else {
492             printf("You win the game if you have a final score greater than %d
                        .\n\n",init_dsum);
493             sleep(2);
494             if (sum1 != 0 && sum1 > init_dsum) {
495                 printf("Player 1 wins the game with %d points! ->
                            BLACKJACK <-\n",sum1);
496             } else {
497                 printf("Player 1 final score is less than dealer's score!\n");
498                 printf("Player 1 loses the game!\n\n");
499             }
500             if (sum2 != 0 && sum2 > init_dsum) {
```

```
501                 printf("Player 2 wins the game with %d points! ->
                        BLACKJACK <-\n",sum2);
502             } else {
503                 printf("Player 2 final score is less than dealer's score!\n");
504                 printf("Player 2 loses the game!\n\n");
505             }
506         }
507     } else {
508         printf("You win the game if you have a final score greater than %d.\n
                \n",init_dsum);
509         sleep(2);
510         if (sum1 != 0 && sum1 > init_dsum) {
511                 printf("Player 1 wins the game with %d points! ->
                        BLACKJACK <-\n\n",sum1);
512             } else if (sum1 > 0) {
513                 printf("Player 1 final score is less than dealer's score!\n");
514                 printf("Player 1 loses the game!\n\n");
515             } else {
516                 printf("Player 1 final score is greater than 21!\n");
517                 printf("Player 1 loses the game!\n\n");
518             }
519         if (sum2 != 0 && sum2 > init_dsum) {
520              printf("Player 2 wins the game with %d points! ->
                     BLACKJACK <-\n\n",sum2);
521             } else if (sum2 > 0) {
522                 printf("Player 2 final score is less than dealer's score!\n");
523                 printf("Player 2 loses the game!\n\n");
524             } else {
525                 printf("Player 2 final score is greater than 21!\n");
526                 printf("Player 2 loses the game!\n\n");
527             }
528     }
529
530     // linux
531     printf("Press ENTER to continue...\n\n");
532     getchar();
533
534     // windows
535     // printf("Press any key to continue...\n\n");
536     // getch();
537
538     return 0;
539 }
```