

```

1  #include "Player.h"
2  #include "Card.h"
3  #include "RatATat.h"
4
5  #define _CRTDBG_MAP_ALLOC
6  #include <crtdbg.h>
7  #ifdef _DEBUG
8  #ifndef DBG_NEW
9  #define DBG_NEW new ( _NORMAL_BLOCK , __FILE__ , __LINE__ )
10 #define new DBG_NEW
11 #endif
12 #endif // _DEBUG
13
14 Player::Player(string name, RatATat* pointer_to_game_ratatat)
15 {
16     this->m_player_name = name;
17     this->m_player_pointer_to_ratatat = pointer_to_game_ratatat;
18     m_player_card_arr = new Card * [m_player_number_of_cards];
19     memset(m_player_card_arr, NULL, m_player_number_of_cards * sizeof (Card*));
20
21     this->m_player_cards_sum = 0;
22     this->m_indx_to_put_a_card_in_player_hand = 0;
23 }
24
25 Player::~Player()
26 {
27     for (int i = 0; i < m_player_number_of_cards; i++)
28         delete m_player_card_arr[i];
29     delete[] m_player_card_arr;
30 }
31
32 ostream& operator<<(ostream& os, const Player& player)
33 {
34     os << player.m_player_name << ": " << endl;
35     os << *player.m_player_card_arr[0] << " , ";
36     os << "hidden card , hidden card , ";
37     os << *player.m_player_card_arr[3] << endl;
38     return os;
39 }
40
41 ///////////////////////////////////////////////////
42 //"Extras":
43
44 string Player::get_player_name()
45 {
46     return this->m_player_name;
47 }
48
49 unsigned int Player::get_player_cards_sum() const
50 {
51     return this->m_player_cards_sum;
52 }

```

```
53
54 int Player::get_player_min_card_indx()
55 {
56     return m_player_min_card_indx;
57 }
58
59 int Player::get_player_max_card_indx()
60 {
61     return m_player_max_card_indx;
62 }
63
64 void Player::print_msg_choose_card_indx() const
65 {
66     cout << "which card do you want want to choose? (0-3)" << endl;
67 }
68
69 void Player::put_card_to_player_hand(Card* card)
70 {
71     m_player_card_arr[m_indx_to_put_a_card_in_player_hand] = card;
72     m_indx_to_put_a_card_in_player_hand++;
73 }
74
75 void Player::calc_sum_of_player_hand()
76 {
77     for (int i = 0; i < m_player_number_of_cards; i++)
78         m_player_cards_sum = m_player_cards_sum + m_player_card_arr[i] - >
79         >get_card_value();
80 }
81
82 void Player::show_player_hand_and_make_it_valid_to_calc_sum()
83 {
84     cout << m_player_name << ": " << endl;
85     for (int i = 0; i < m_player_number_of_cards; i++)
86     {
87         cout << *m_player_card_arr[i] << " , ";
88         //here need to make hand valid to calc sum , then show hand again:
89         while (m_player_card_arr[i]->is_special_card())
90         {
91             m_player_pointer_to_ratatat->throw_card_to_discard_pile
92             (m_player_card_arr[i]);
93             m_player_card_arr[i] = m_player_pointer_to_ratatat -
94             >get_card_from_unused_pile();
95         }
96     }
97     cout << endl;
98     for (int i = 0; i < m_player_number_of_cards; i++)
99         cout << *m_player_card_arr[i] << " , ";
100     cout << endl;
101 }
102
103 Card* Player::get_card_pointer_by_indx(int indx)
104 {
105     return m_player_card_arr[indx];
106 }
```

```
103 }
104
105 //void Player::show_card_by_indx(int indx)
106 //{
107 //    cout << *m_player_card_arr[indx];
108 //}
109
110 void Player::replace_cards(int indx_of_card_to_replace, Card*&
    card_b_pointer)
111 {
112     Card* tmp_card_pointer = m_player_card_arr[indx_of_card_to_replace];
113     m_player_card_arr[indx_of_card_to_replace] = card_b_pointer;
114     card_b_pointer = tmp_card_pointer;
115     return;
116 }
117
118 ///////////////////////////////////////////////////
119 //class AI_player:
120
121 int AI_player::ChooseOption_from_card_menu(Card& currentCard)
122 {
123     int max_option = currentCard.get_card_action_menu_max_num();
124     int rand_option = 1 + rand() % max_option;
125     return rand_option;
126 }
127
128 int AI_player::choose_card_indx() //for play cards
129 {
130     print_msg_choose_card_indx();
131     int rand_option = rand() % m_player_max_card_indx;
132     cout << rand_option << endl;
133     return rand_option;
134 }
135
136 int AI_player::choose_player(Player** players_arr, int curr_player) //for
swap!
137 {
138     cout << "with which player do you want to swap card with?" << endl;
139     int player_answer_which_player_to_swap_with = rand() %
    m_player_pointer_to_ratatat->get_number_of_players();
140     while (player_answer_which_player_to_swap_with == curr_player)
141         player_answer_which_player_to_swap_with = rand() %
    m_player_pointer_to_ratatat->get_number_of_players();
142     cout << player_answer_which_player_to_swap_with << endl;
143     return player_answer_which_player_to_swap_with;
144 }
145
146 int AI_player::player_choice_from_what_pile_to_pick(int number_of_option)
147 {
148     int ans(1 + rand() % number_of_option);
149     cout << ans << endl;
```

```

150     return ans;
151 }
152
153 int AI_player::player_answer_to_call_ratatat(int number_of_option)
154 {
155     int ans = (1 + rand() % number_of_option);
156     cout << ans << endl;
157     return ans;
158 }
159
160 //////////////////////////////////////////////////
161 //class Human_player:
162
163 int Human_player::ChooseOption_from_card_menu(Card& currentCard)
164 {
165     int choice;
166     int max_option = currentCard.get_card_action_menu_max_num();
167     cin >> choice;
168     while (choice < 1 || choice > max_option)
169         cin >> choice;
170     return choice;
171 }
172
173 int Human_player::choose_card_indx()
174 {
175     int player_answer_which_card_to_choose;
176     print_msg_choose_card_indx();
177     cin >> player_answer_which_card_to_choose;
178     while (player_answer_which_card_to_choose <
179           Player::get_player_min_card_indx() ||
180           player_answer_which_card_to_choose >
181           Player::get_player_max_card_indx())
182         cin >> player_answer_which_card_to_choose;
183     return player_answer_which_card_to_choose;
184 }
185
186 int Human_player::choose_player(Player** players_arr, int curr_player)
187 {
188     int player_answer_which_player_to_swap_with;
189     int number_of_players = m_player_pointer_to_ratatat -
190         >get_number_of_players();
191     cout << "with which player do you want to swap card with?" << endl;
192     int i;
193     for (i = 0; i < number_of_players; i++)
194         cout << i << " ) " << players_arr[i]->get_player_name() << " , ";
195     cout << endl;
196     cin >> player_answer_which_player_to_swap_with;
197     while (player_answer_which_player_to_swap_with < 0 ||
198           player_answer_which_player_to_swap_with >= number_of_players ||
199           player_answer_which_player_to_swap_with == curr_player)
200         cin >> player_answer_which_player_to_swap_with;
201 }

```

```
196     return player_answer_which_player_to_swap_with;
197 }
198
199 int Human_player::player_choice_from_what_pile_to_pick(int           ↗
    number_of_option)
200 {
201     int answer;
202     cin >> answer;
203     while (answer < 1 || answer > number_of_option)
204         cin >> answer;
205     return answer;
206 }
207
208 int Human_player::player_answer_to_call_ratatat(int number_of_option)
209 {
210     int answer;
211     cin >> answer;
212     while (answer < 1 || answer > number_of_option)
213         cin >> answer;
214     return answer;
215 }
216
```