

```
1. pub fn swap_colors(  
2.   colors: List(String),  
3.   game_subject: Subject(GameActorMessage),  
4.   state: GameActorState,  
5. ) {  
6.   let indexed_colors =  
7.     colors  
8.     |> list.index_map(fn(color, i) { #(i, color) })  
9.  
10.   //map participants to new participants with updated colors & update players' state  
11.   let new_participants =  
12.     state.participants  
13.     |> list.map(fn(participant) {  
14.       let assert Ok(color) =  
15.         indexed_colors  
16.         |> list.find(fn(color) { color.0 == { participant.0 }.number - 1 })  
17.       let new_player = Player(..participant.0, color: color.1)  
18.       process.send(participant.1, UpdatePlayerState(new_player))  
19.       #(new_player, participant.1)  
20.     })  
21.  
22.   let new_state =  
23.     GameActorState(..state, participants: new_participants, used_colors: colors)  
24.   //send a message to every participant, updating the page  
25.   list.each(new_state.participants, fn(participant) {  
26.     process.send(  
27.       participant.1,  
28.       SendToClient(  
29.         player_container(new_state, game_subject) |> element.to_string,  
30.       ),  
31.     )  
32.   })  
33.  
34.   new_state  
35. }
```