

```

swapColors(colors, game_subject) -> {
  let indexed_colors =
    colors
  |> list.index_map(fn(color, i) { #(i, color) })

  //map participants to new participants with updated colors & update players' state
  let new_participants =
    state.participants
  |> list.map(fn(participant) {
    let assert Ok(color) =
      indexed_colors
    |> list.find(fn(color) { color.0 == { participant.0 }.number - 1 })
    let new_player = Player(..participant.0, color: color.1)
    process.send(participant.1, UpdatePlayerState(new_player))
    #(new_player, participant.1)
  })

  let new_state =
    GameActorState(
      ..state,
      participants: new_participants,
      used_colors: colors,
    )
  //send a message to every participant, updating the page
  list.each(new_state.participants, fn(participant) {
    process.send(
      participant.1,
      SendToClient(
        player_container(new_state, game_subject) |> element.to_string,
      ),
    )
  })
  new_state |> actor.continue
}

```