Virtual Deck

a mobile application to replace your deck of cards

developed by Joseph Faraci and Trevor Buchan

For SteelHacks 2016

Goals and Achievements

- We set out to create an app that would closely emulate a physical, standard pack of playing cards
- We successfully designed a way to do this, but were unable to fully implement the app.
- We were able to create several sample classes to describe how all of the objects would be instantiated and interact with each other.

Overview

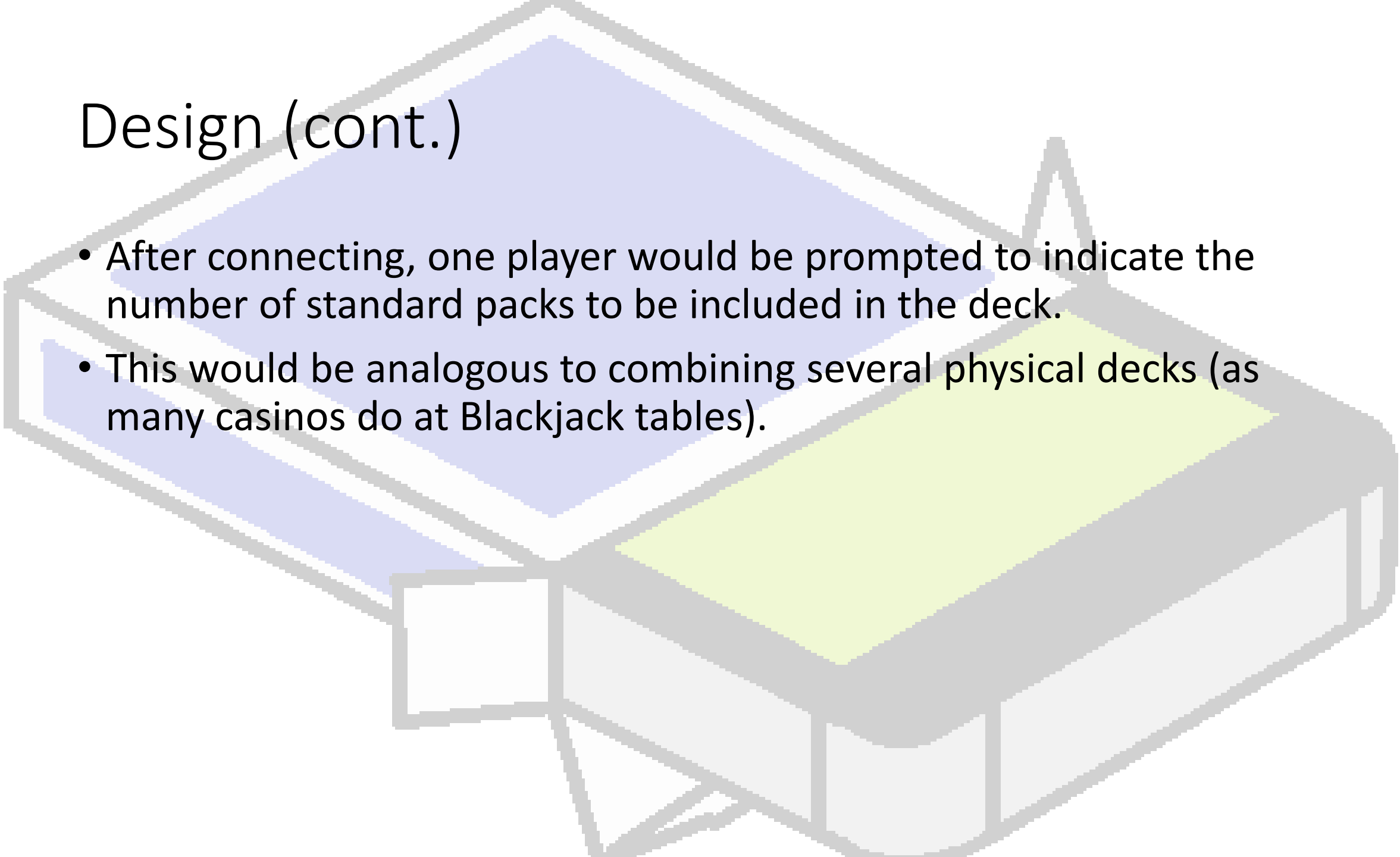
- The Virtual Deck is extremely generalized, so that almost any game that can be played with physical cards can be played using the app.
- THIS WAS NOT INTENDED TO BE AN ONLINE GAME
- We did not intend to allow players to connect to others via the internet.
- Furthermore, players would only be able to successfully play a game while in the same room with each other, just like they would with a physical pack of cards.

Design

- The app would start by prompting users to start a game.
- It would then search for other Bluetooth enabled mobile phones to connect to.
- The players would select each other’s phones and allow connection.

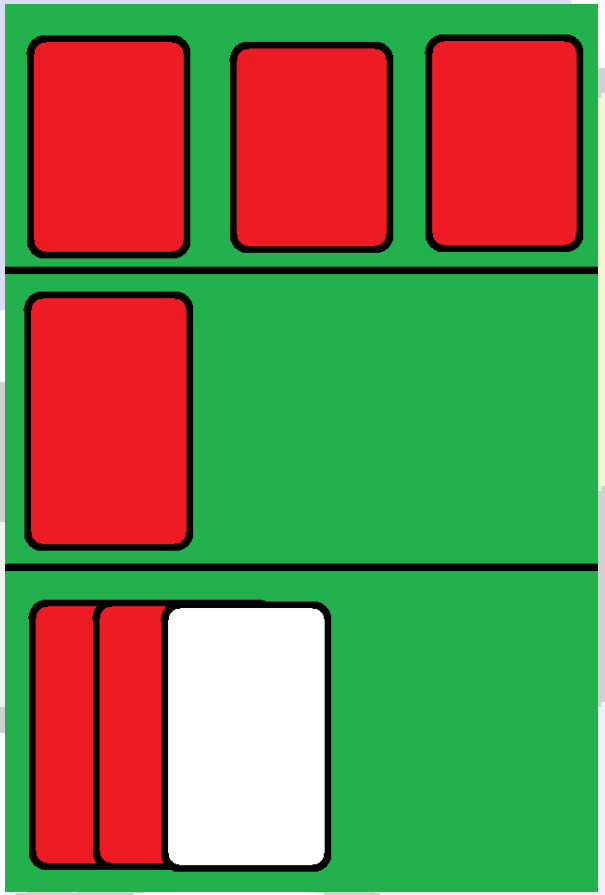
Design (cont.)

- After connecting, one player would be prompted to indicate the number of standard packs to be included in the deck.
- This would be analogous to combining several physical decks (as many casinos do at Blackjack tables).



Design (cont.)

- The players would then be presented with a screen with three sections, similar to this:

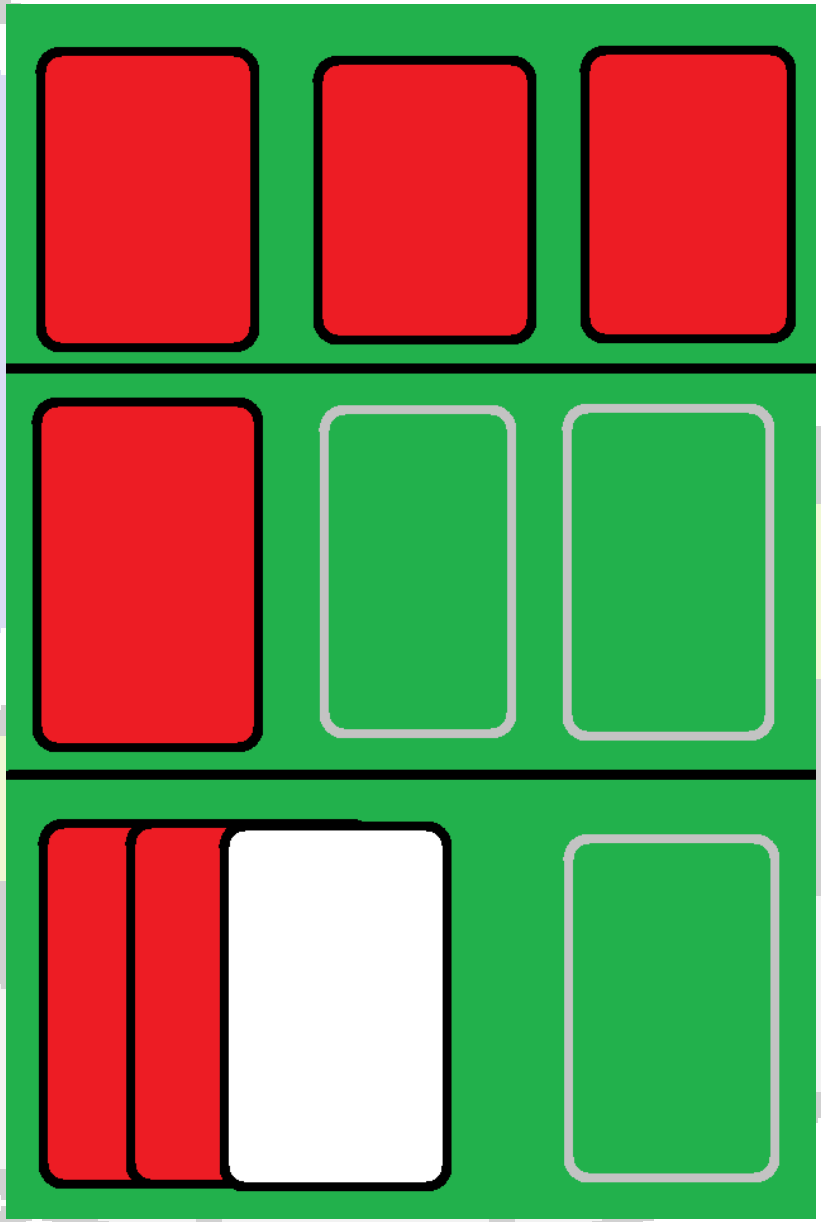


Design (cont.)

Other players' hands

The deck and discard piles

The player's hand and played cards

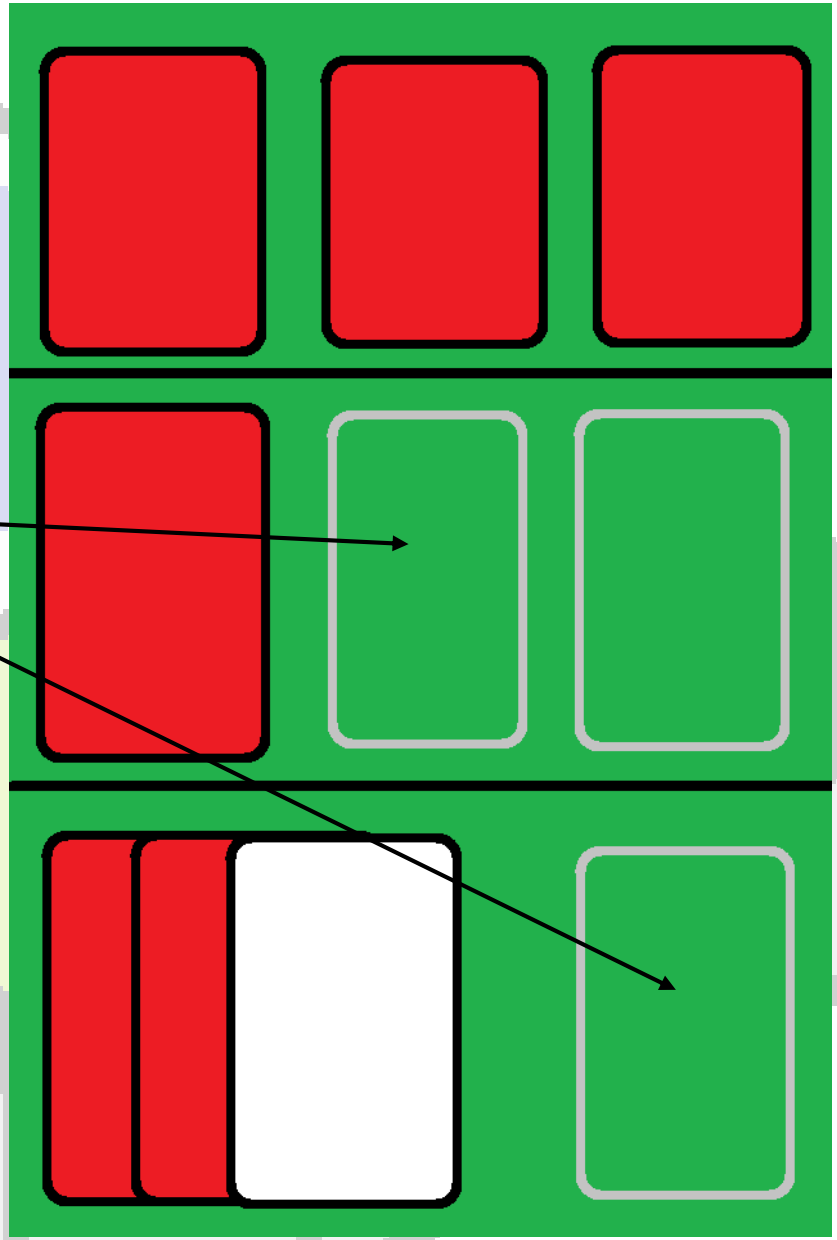


Design (cont.)

- Each player would be able to see the deck, but not the cards in it. Initially, this is the only field with any cards in it.
- Each player would be able to see their own hand, and well as any cards that they would play (such as laying down a hand for poker).
- The hand of the other players would also be represented, but the cards would not be able to be seen by the player.
- The other players' section would also be used to display cards that the other players "laid down."

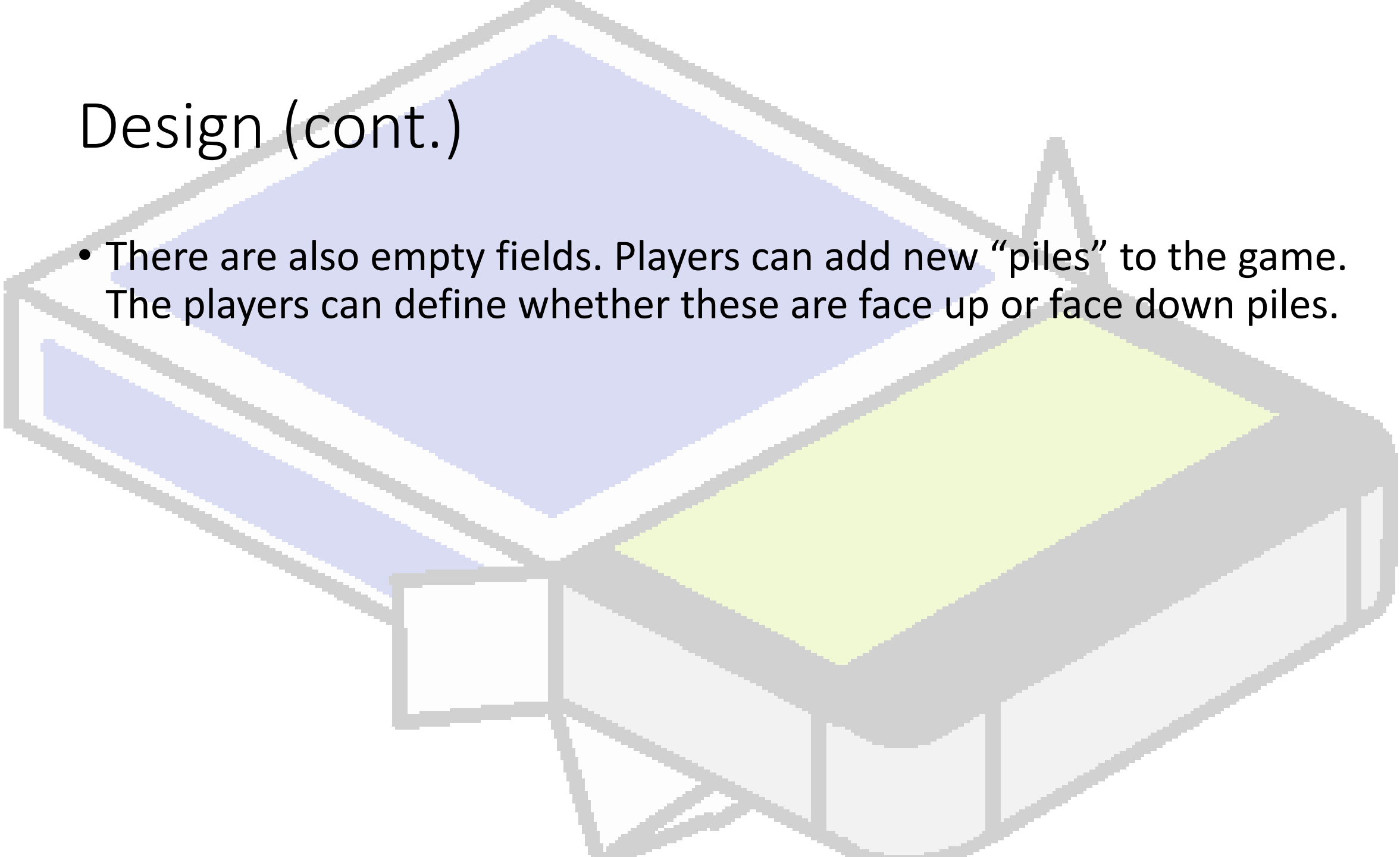
Design (cont.)

Option to add more piles



Design (cont.)

- There are also empty fields. Players can add new “piles” to the game. The players can define whether these are face up or face down piles.



Design (cont.)

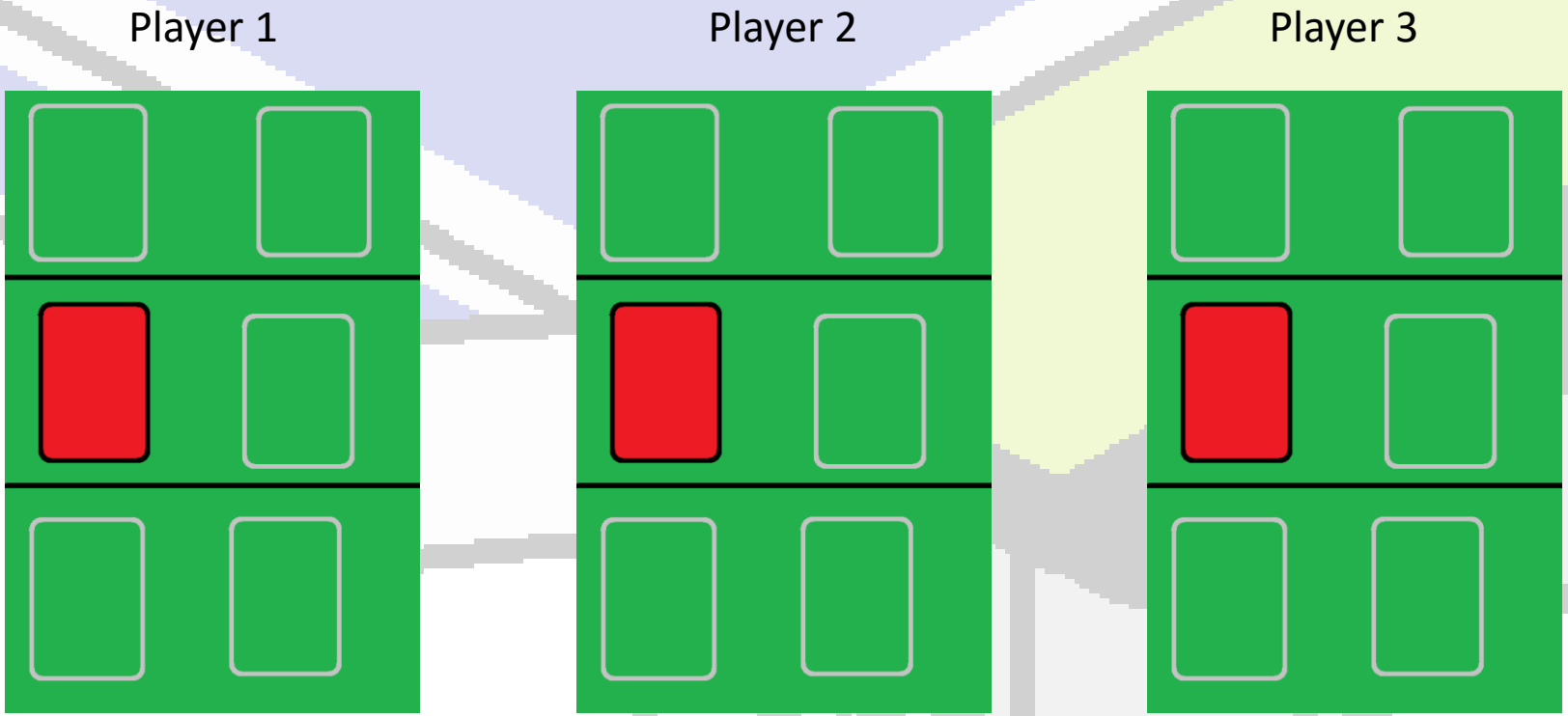
- Every player is able to draw cards from the deck, give cards to other players, and place cards in the discard pile. This would be achieved by simply dragging and dropping cards.
- In order to allow use in as many games as possible, these movements (and similar ones) would be the only rules enforced by the app. No turns or rounds are defined.
- But what is keeping players from cheating?

Design (cont.)

- Every time that a card is moved, all other players will be notified. It is up to the other players, not the app, to enforce the rules of the game.
- Note that this is exactly how physical card games work. Nothing is stopping players from drawing extra cards, except for the watchful eyes of their competitors.
- However, Virtual Deck does have the added benefit of keeping players from sneaking peeks at other hands.

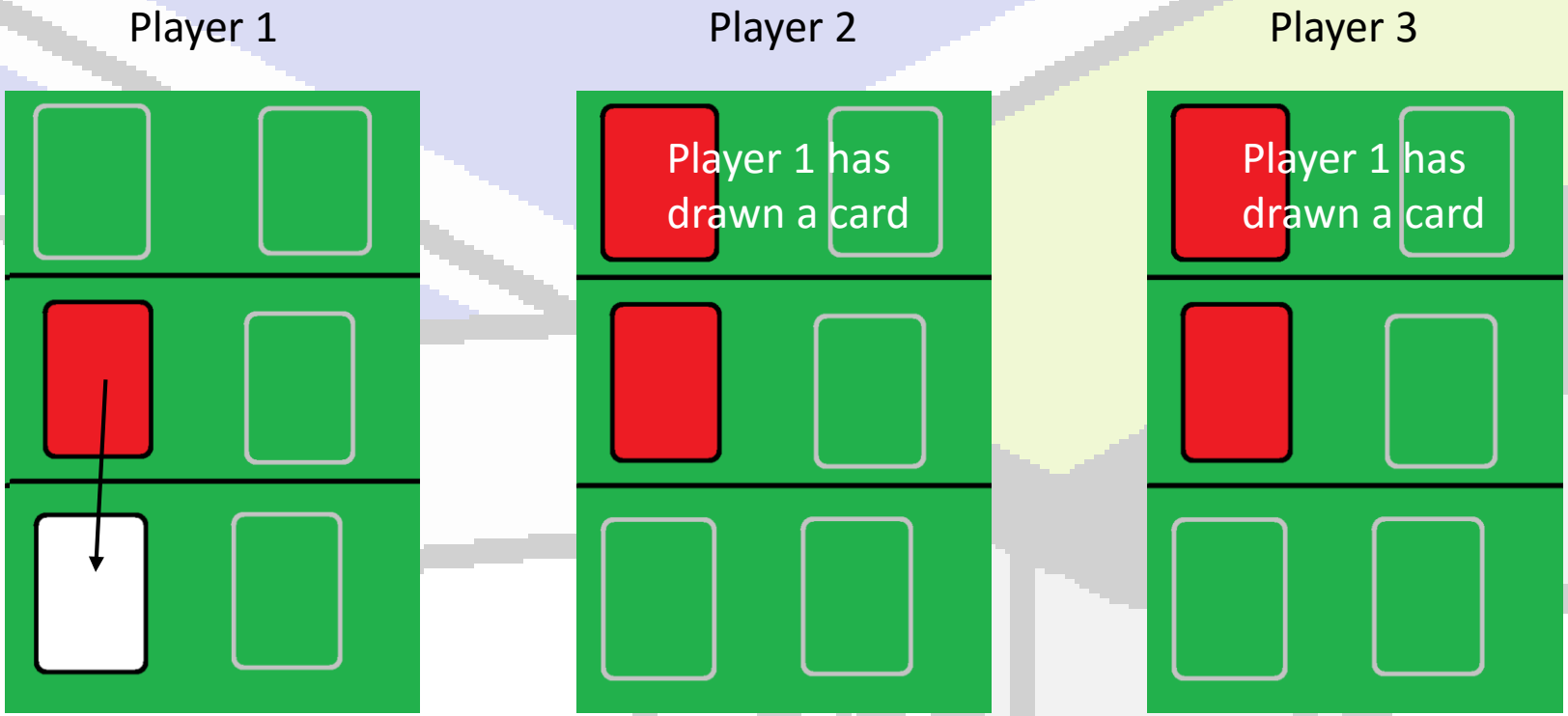
Example

- Take this sample game of Go Fish with three players for example.
- The game starts, and only the deck is displayed



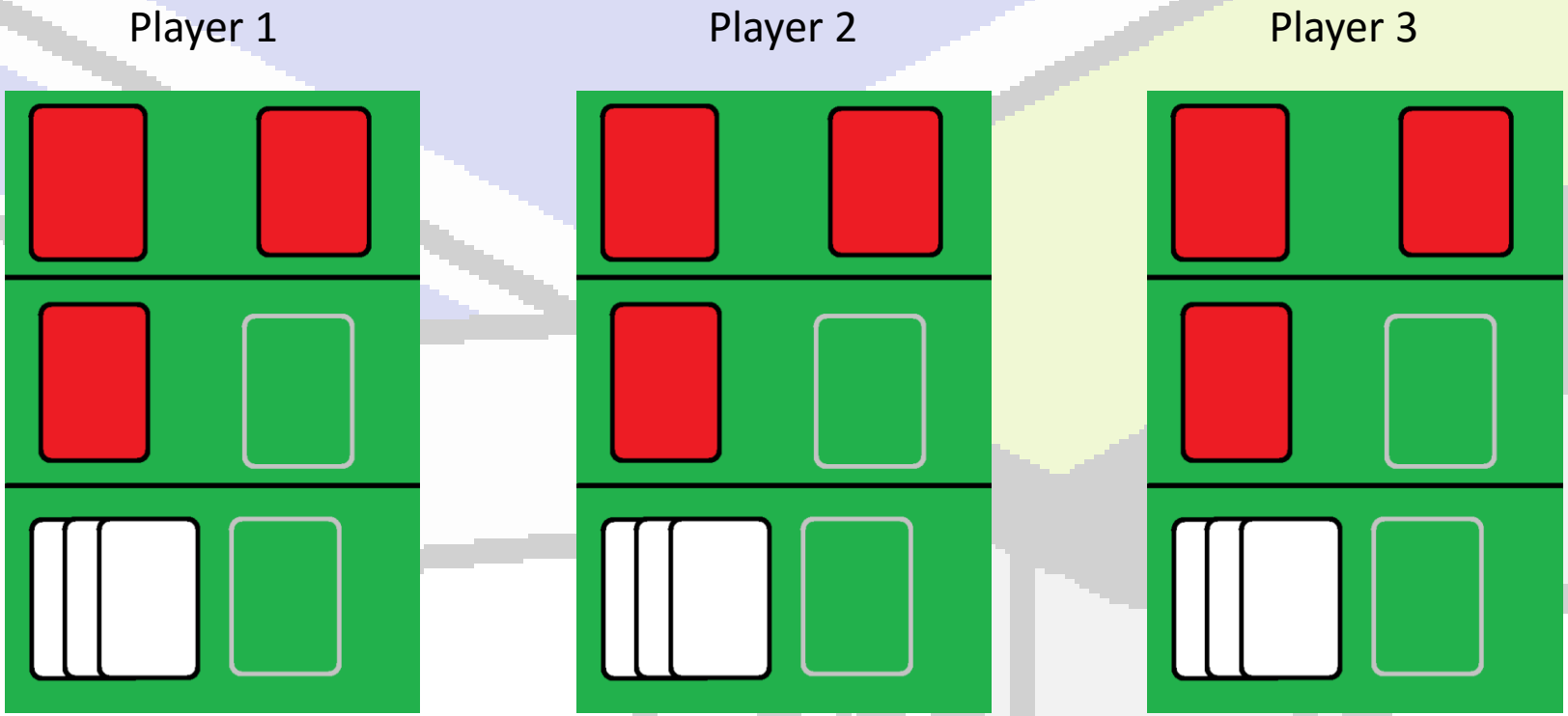
Example (cont.)

- Player one draws a card.
- The other players are notified.



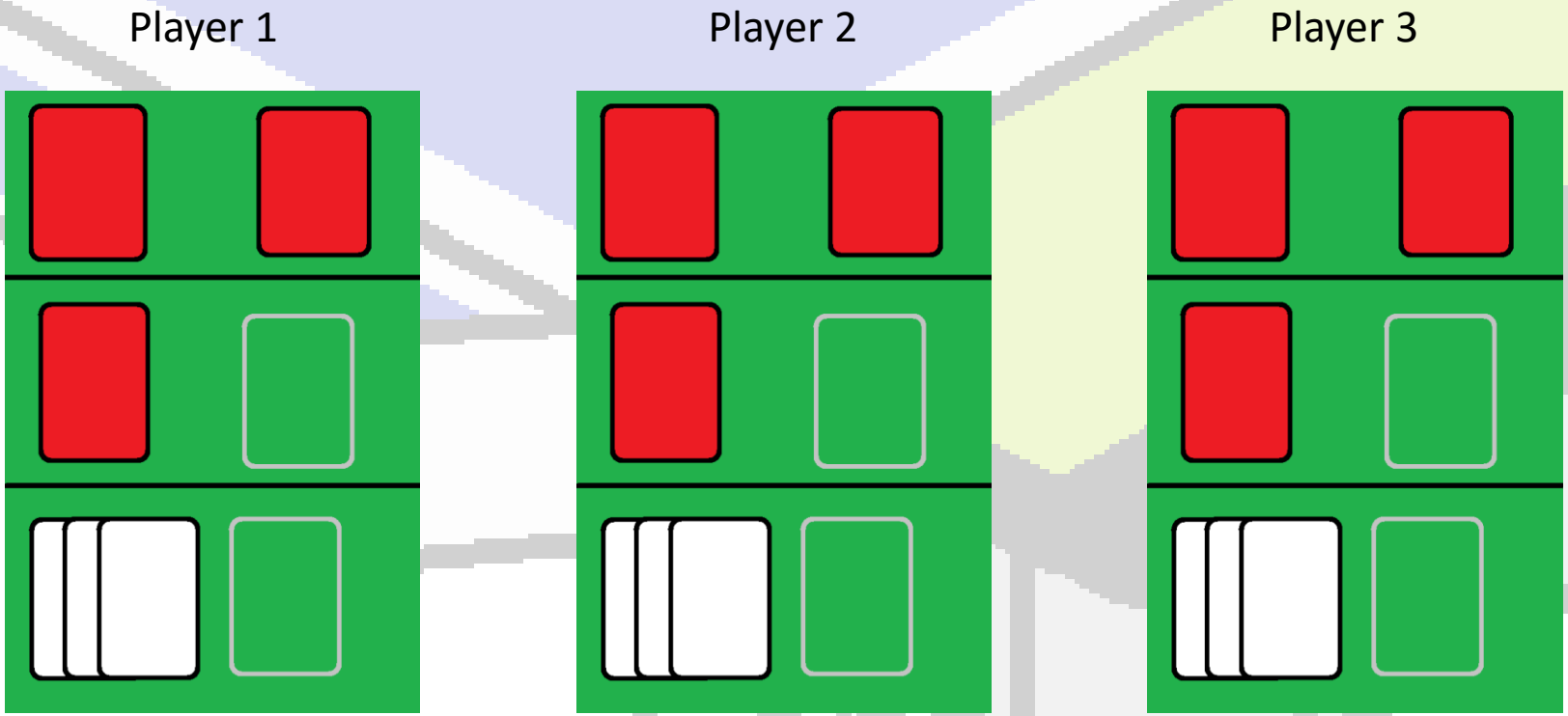
Example (cont.)

- This continues until each player has drawn the correct number of cards.



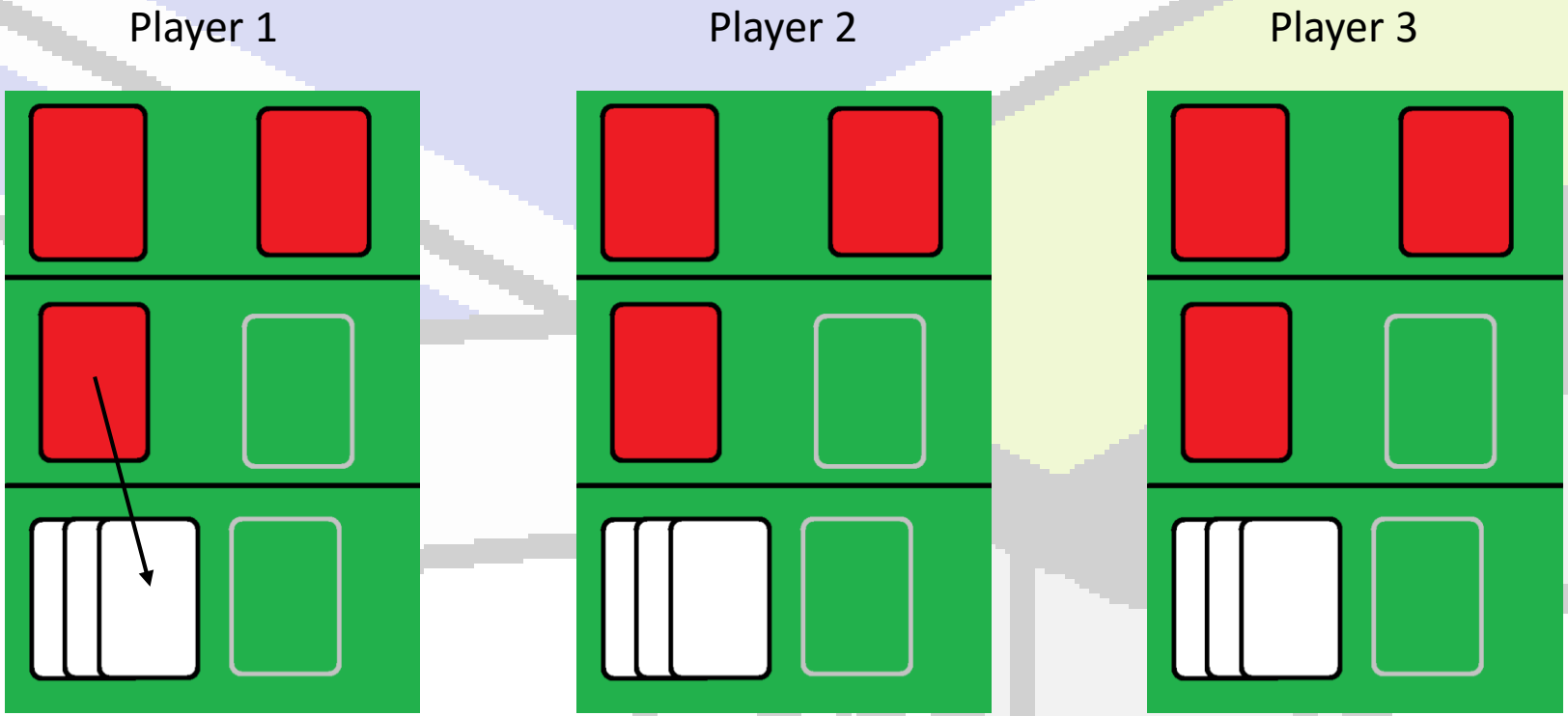
Example (cont.)

- Suppose Player 1 now verbally asks Player 2 if they have any 3s.



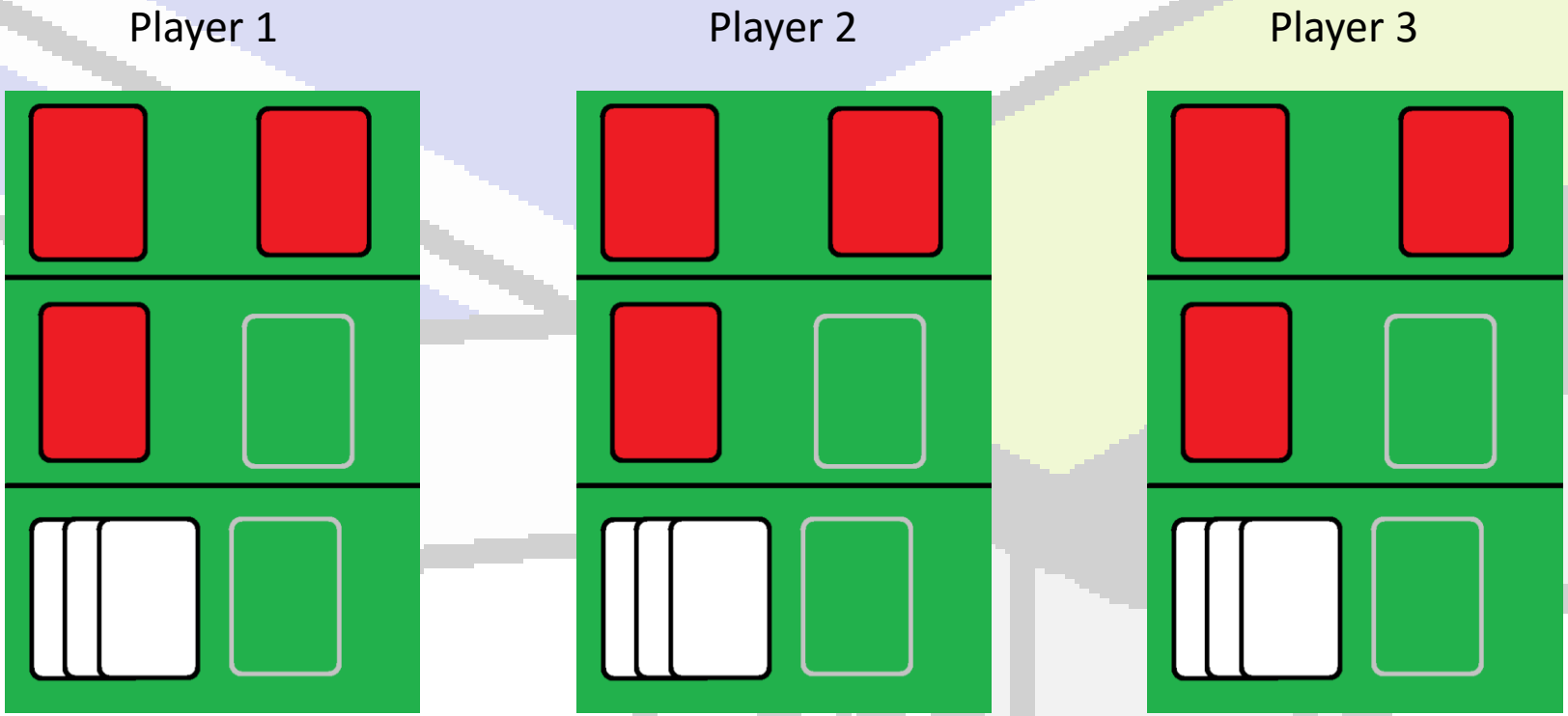
Example (cont.)

- Suppose Player 2 responds “Go Fish”.
- Player 1 draws from the deck.



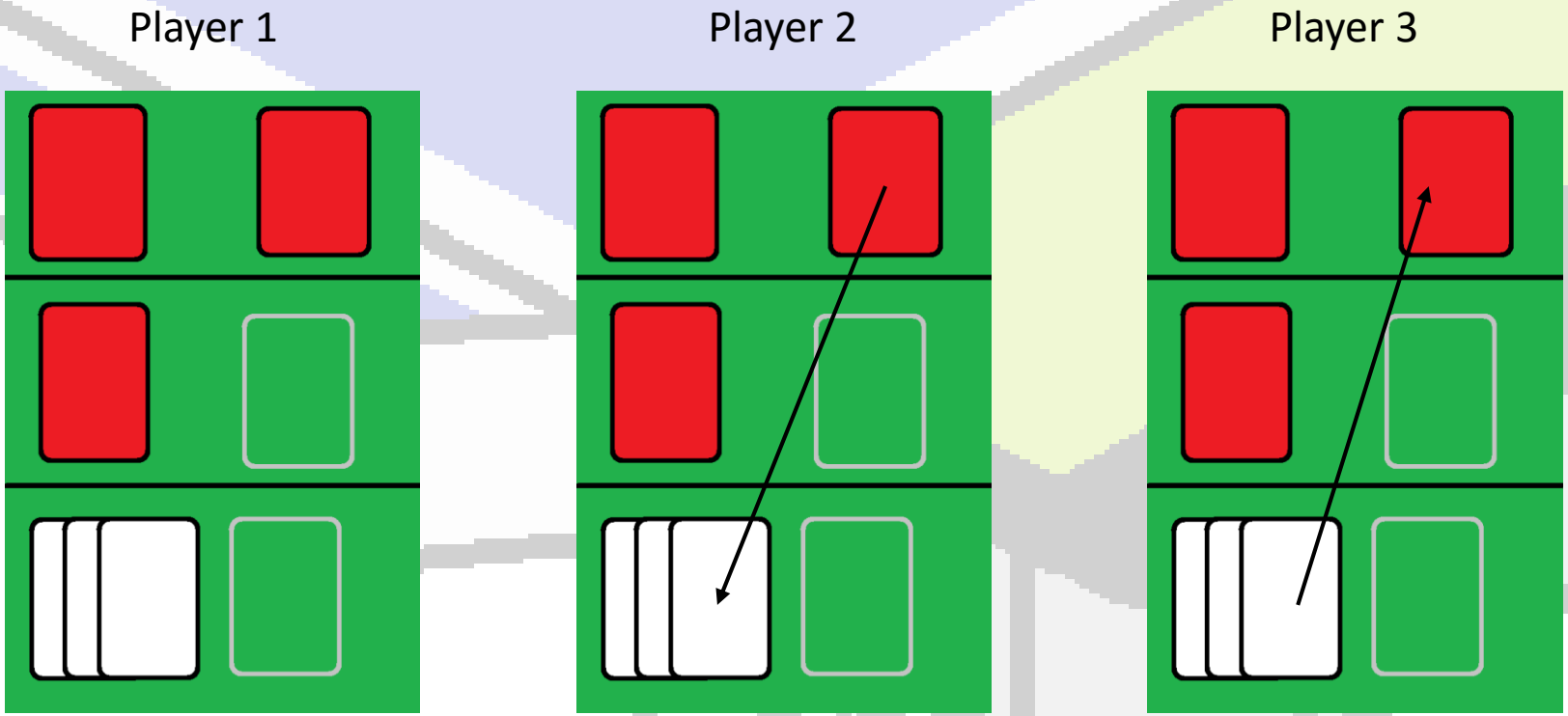
Example (cont.)

- Suppose Player 2 now verbally asks Player 3 if they have any 4s.



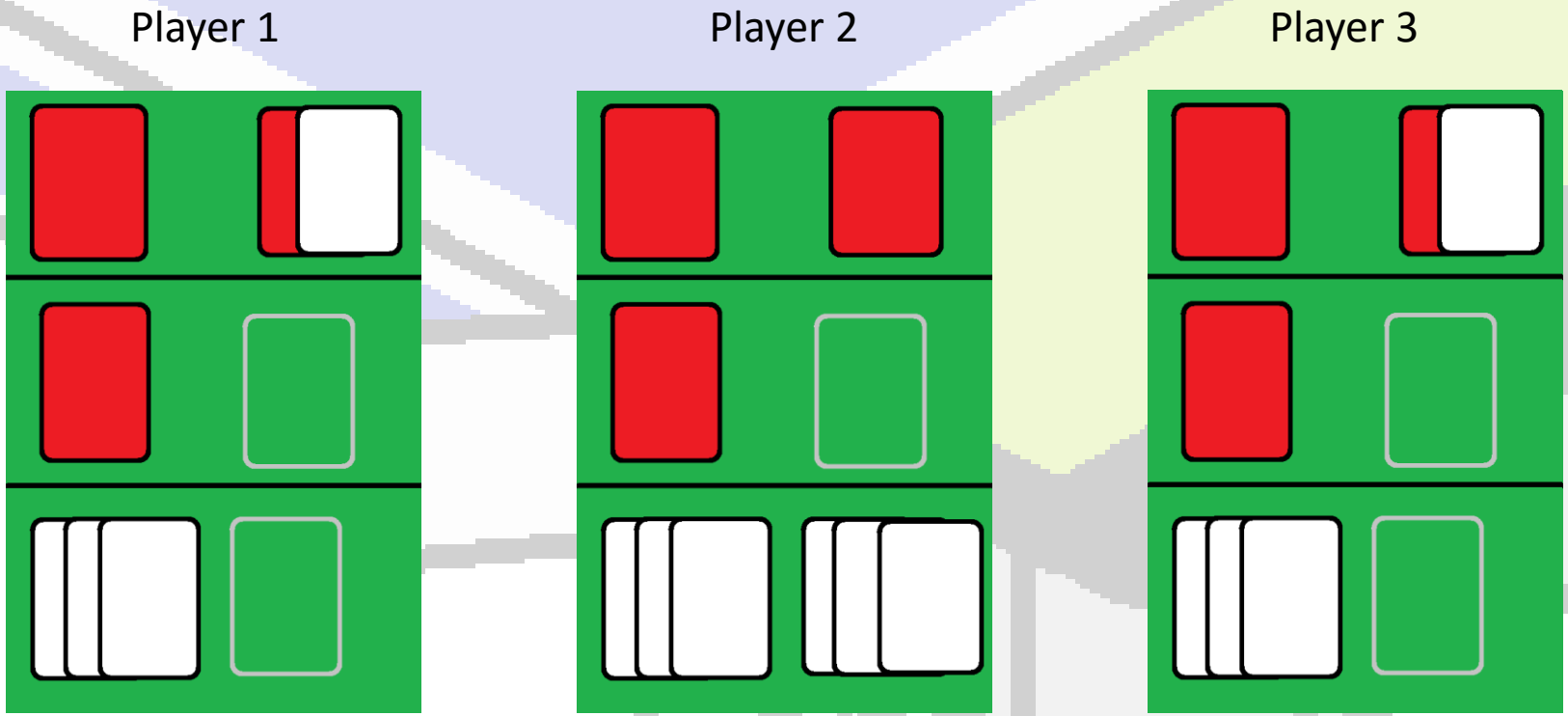
Example (cont.)

- Player 3 has two 4s, and moves them to Player 2's hand.



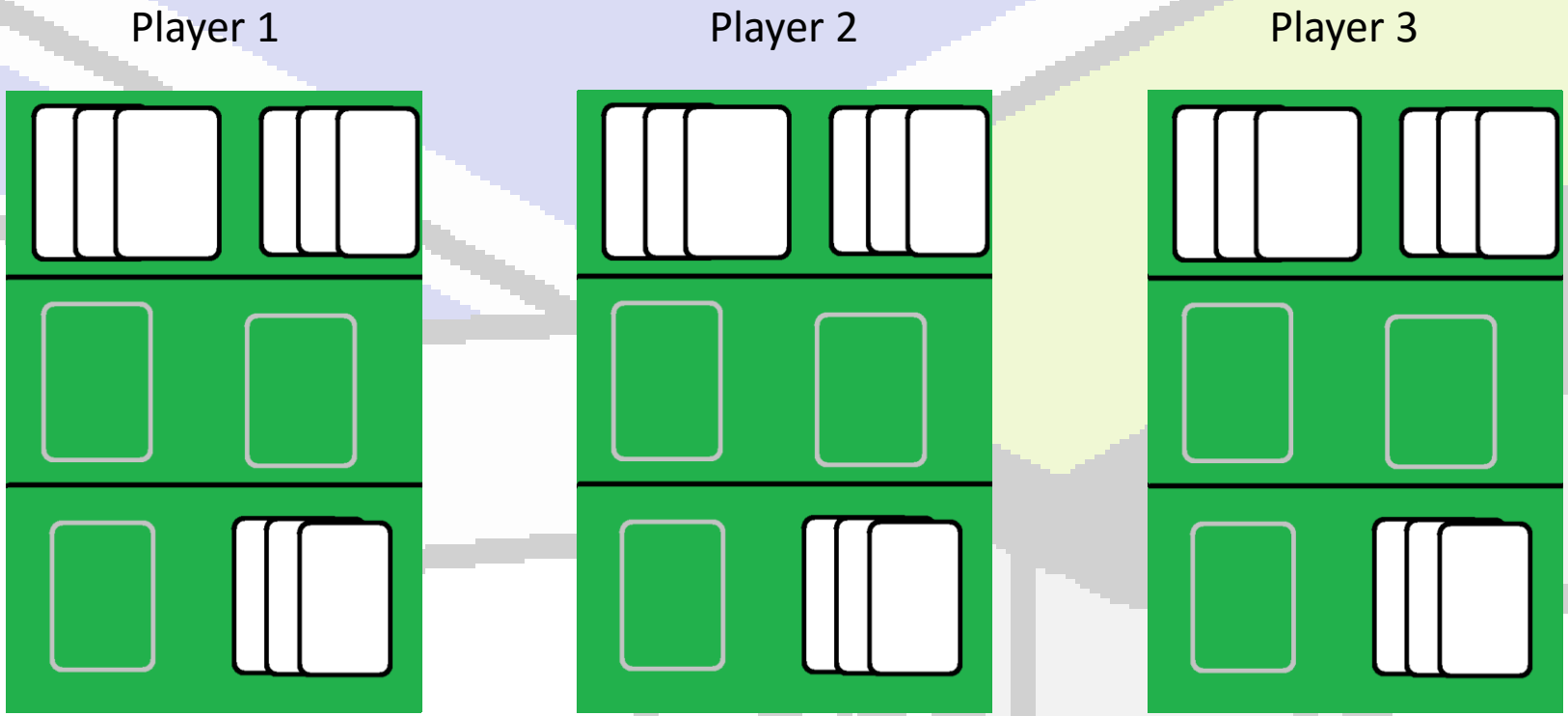
Example (cont.)

- Player 2 had two 4s, so they now hold all four. They can create a new face up pile in their own field to move these cards to.



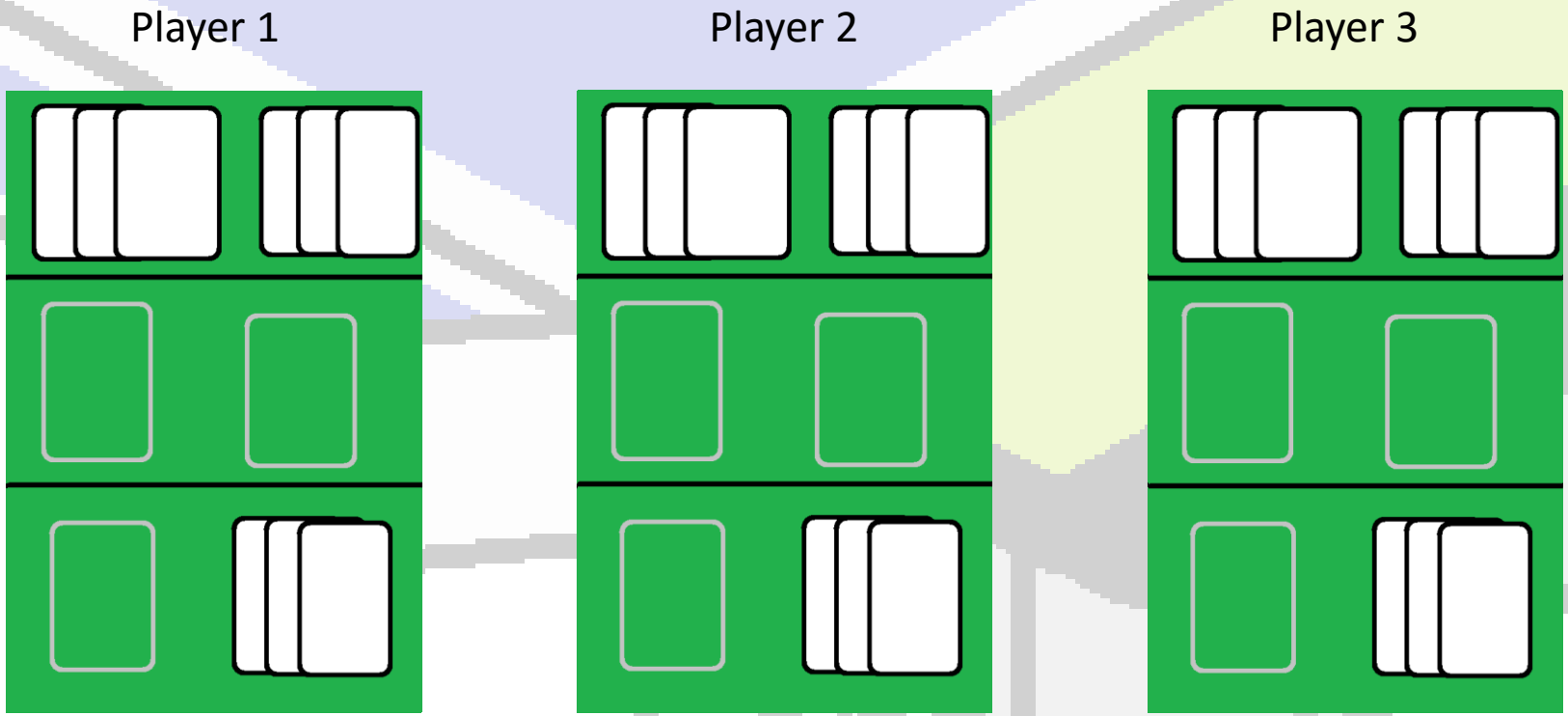
Example (cont.)

- After several rounds, the deck is empty, and all books have been completed.



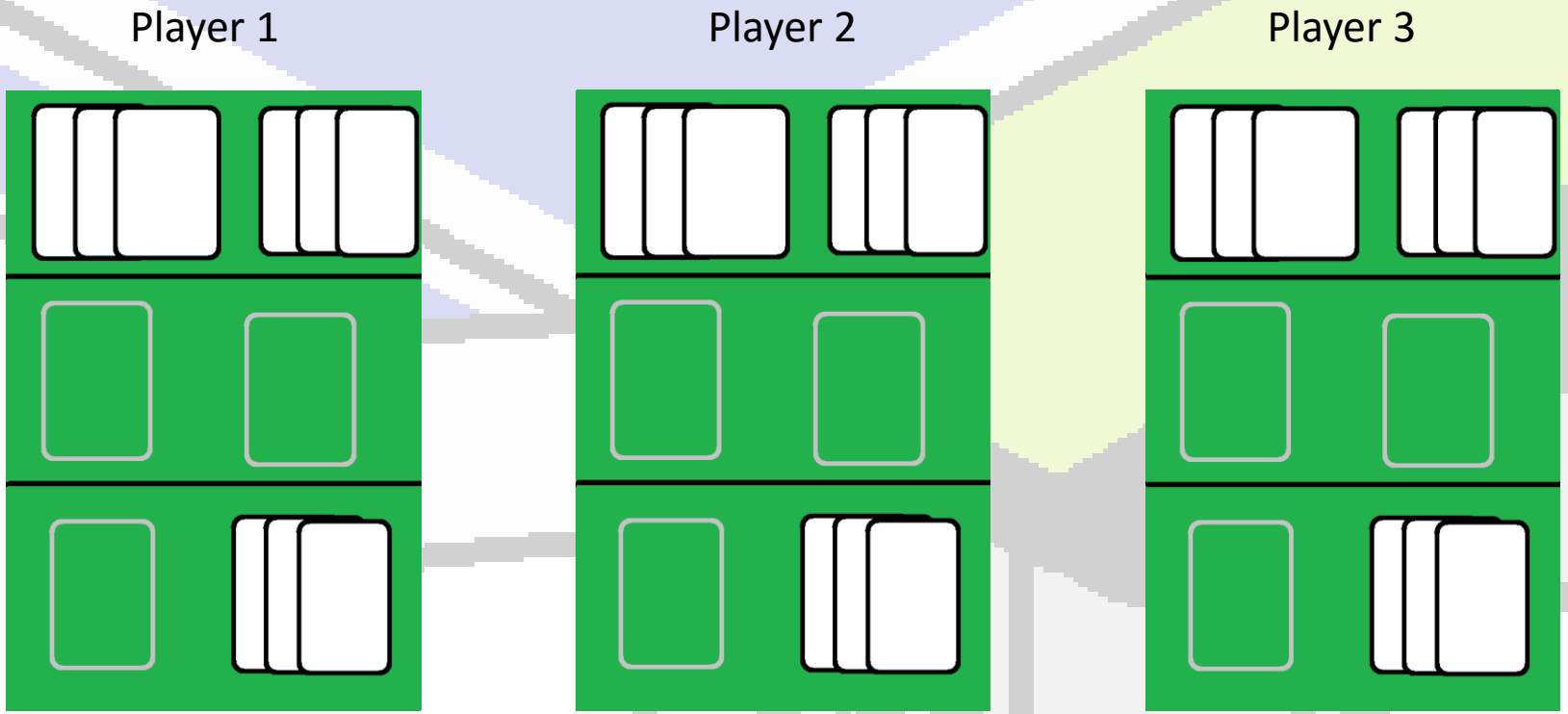
Example (cont.)

- The players count who has the most completed groups of ranks, and decide the winner.



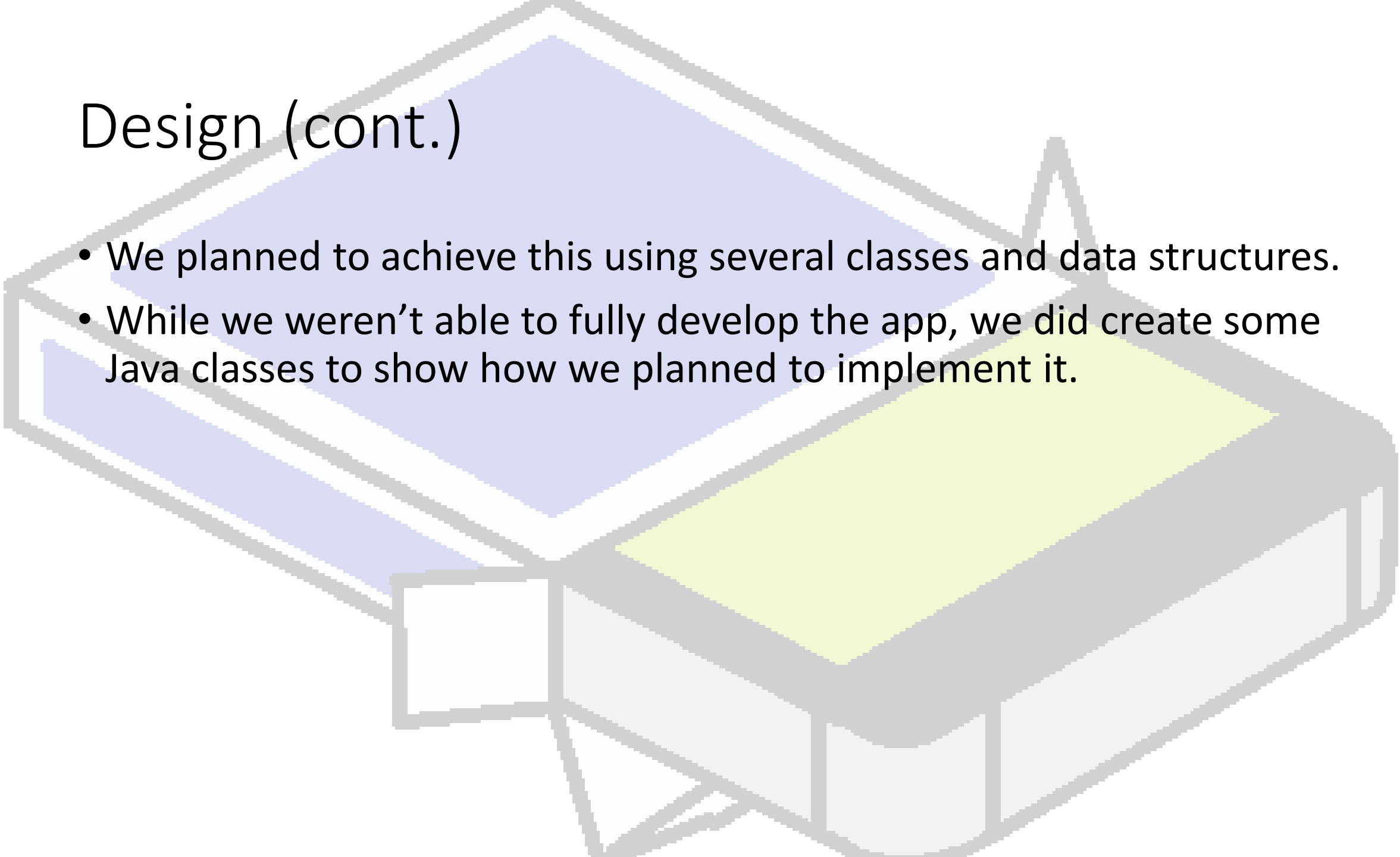
Example (cont.)

- Notice that the app did not keep track of how many full sets each player had. It also did not mediate communication between players.



Design (cont.)

- We planned to achieve this using several classes and data structures.
- While we weren't able to fully develop the app, we did create some Java classes to show how we planned to implement it.



Classes

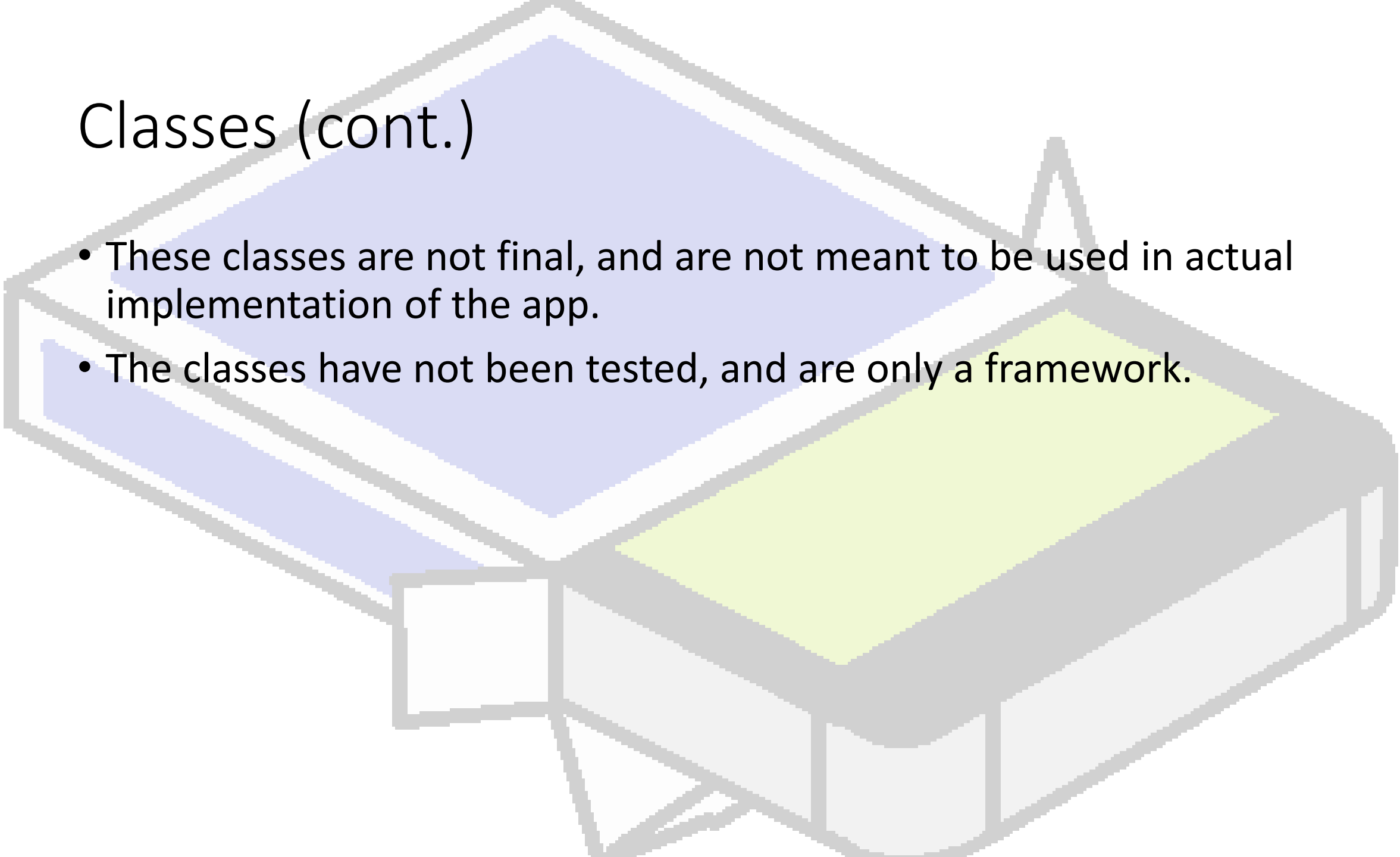
- Card
  - Cards are the main objects in our design. We created a class to represent them. This class does not include visual assets, and is not meant to actually be implemented. It is only the basic framework for what we hoped to use in the finished product.
  - The class Card includes the following variables:
    - int rank; //the face value of the card
    - int suit; //the suit of the card, represented by the integers from 0 to 3
    - boolean faceUp; //determines whether the card is face up or down

Classes (cont.)

- Pile
  - Pile is a data structure that holds Card objects in an array list. It is used to represent decks, hands, discard piles, etc.
  - It includes one instant variable:
    - ArrayList<Card> pile; //the Array List that holds Card objects
- Player
  - Player objects hold each player's ID and hands.
  - It includes the variables:
    - String name; //The player's ID in a string form. "Player 1", "Player 2", etc.
    - int id; //The player's ID. Each player is assigned an integer ID from 0 and up.
    - ArrayList<Pile> hand; //An arraylist to hold every pile in the player's hand

Classes (cont.)

- These classes are not final, and are not meant to be used in actual implementation of the app.
- The classes have not been tested, and are only a framework.



Moving Foward

- Obviously, this was not finished during SteelHacks 2016, and will not be finished nor attempted at later hackathons.
- However, both of us have expressed an interest in revisiting the project in the future for our own enrichment.
- In addition to the implementation of what we have described in this presentation, we would like to add other features, such as different methods of shuffling the deck to emulate shuffling techniques used by humans, some advanced card movements (rapid passing of cards for games like Spoons, for example), etc.