Eric Gonzalez peer reviewing Andy Newton's solitaire code

Overall, I think that the code is structured well and is understandable to the reader. The functions are clearly named, and the variable names themselves describe their purpose. Additionally, there are plenty of comments describing newly created data structures with detail on what their job is. The code follows a logical structure and has files that describe the various data structures and their attributes. Also, the code follows conventions with indentation, naming, and commenting, making the code readable and clean. I think this was the biggest factor in making the code understandable to me, because there were natural line breaks for different parts of a function as well as line breaks between functions. I think that the code looks great and doesn't need much to improve upon its already great functionality and readability, however I do have a few nitpick suggestions that could be considered to try to make the code even more readable and concise. The first suggestion is to cut down on the total number of file names by taking out magic.lua and conf.lua. Both of these files have few lines, and I think it would make your overall project look more approachable and readable if you just placed these few lines inside of another file. Additionally, I think there might be an opportunity to cut down on the number of lines you have to write for deck.lua and draw_pile.lua by reusing the logic for tableau.lua. All of these card stacks are similar in that they have a set position, and cards can be removed or added to any of these stacks. So if you make the deck and draw piles be tableaus, then you could just have your deck.lua and draw_pile.lua functions override the tableau.lua functions when necessary. Though that's just a potential optimization and not necessary for this assignment, just a stylistic choice that I think would make the code feel more optimized.