NEEDED OBJECTS FOR BLACK JACK TEAM PROJECT

MenuGUI

- The MenuGUI object is the GUI that the user will first see when running the program. It will give the user some options as to how to proceed whether that be playing the game, customizing the game options, or a help page describing the rules.

GameGUI

- The GameGUI object is the GUI that the user will see when playing Blackjack. This will replace MenuGUI if the user selects “Play” in MenuGUI. The GUI will display everything needed for a game of Blackjack. The players and their cards, the dealer, options for the player on how to proceed with their current hand, etc.

Game

- The Game object is responsible for the actual game of Blackjack. It will have methods to be called from GameGUI such as hit(), stand(), etc. It will be responsible for keeping track of all the players and their information and keeping the game going.

Player

- Player is an abstract class that will be inherited by PlayerAI and HumanPlayer.
- Player objects will have information stored about the player such as the player’s current amount of chips and the cards they have in their hand. It will also have methods such as hit(), stand(), double(), etc.

PlayerAI - The PlayerAI object will be used for the extra players in the game. The game will only have single player capabilities so having AI players will give the illusion that the player is playing with others. The AI's will play with the basic Blackjack strategy that we will be basing off of the Blackjack player's handbook. The algorithm will run whenever it is an AI player's turn and will return the next best move to play for the AI to the GameGUI which will then call the corresponding method in the Game object, which will then call the corresponding method in Player.

HumanPlayer - The HumanPlayer object is the same as PlayerAI minus the ability to get the next best move. Instead, the moves done will be selected by the user, aka the human. They will be selected from the GameGUI which will then call the corresponding method in the Game object, which will then call the corresponding method in Player.

CardDeck

- The CardDeck object will contain an arraylist of Card objects representing a "deck". CardDeck will have the capabilities of simulating multiple decks at once, as well as randomly getting a card from the deck.

Card

- The Card object is simply the representation of a playing card. It will hold information about it such as the suit and value. It will also have an image of the card so that we may use it in the GUI.