Abstract Game

+ GameName : String
+GameDeck : Deck
+NumberOfPlayers : int

-setRules() : void
-excludeCards(int[],Deck[]) : int[]
-setDeckValues(Deck) : void
+getDeckValues() : int
+getNumberOfPlayers(int) : int
-setDeckSize(int) : void
+getDeckSize() : int
+getGameName(S) : String
-message() : void
+playerTurn(int player) : int
+discard(Card) : Card
getDiscards() : ArrayList<Card>

NOTE: A player class will need to be added to this diagram if we plan on going through with creating "playable" card games.

△
Extends

Game Driver

-------------Extra fields to be determined

------------To be determined

parent

child

+Deck

+DeckSize : int
+Shuffle : boolean

+shuffle(ArrayList<Card>deck) : ArrayList<Card>
+peek(Card topCard) : Card
+setDeckSize() : void

parent

child

child

Card

-suite : String
-positionValue : int
-color : Color or String
-faceUp : boolean
-cardImage : Image

+revealCard(int) : int
+getColor() : Color/String
+getSuite() : String
+isFaceUp() : boolean
+getPositionValue() : int
+viewImage() : void

Hand

-hand : ArrayList<Card>
-handSize : int

+draw(Deck) : Card
+play(Card) : Card
+getHandSize() : int
+getHand() : ArrayList<Card>