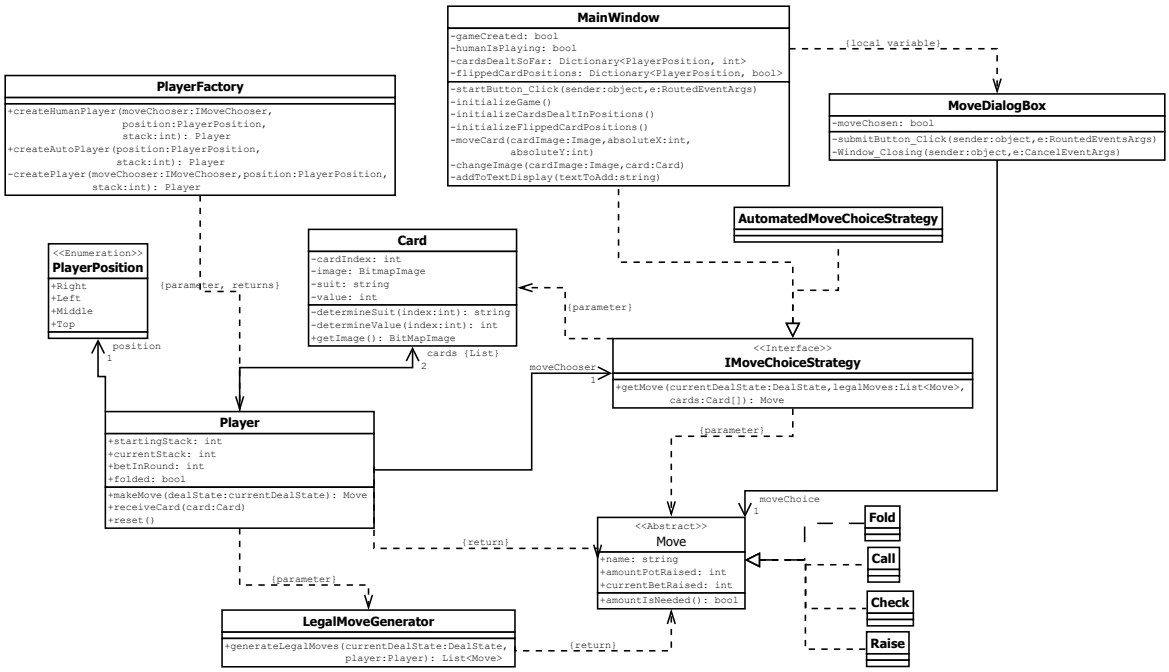
MainWindow
-gameCreated: bool
-humanIsPlaying: bool
-cardsDealtSoFar: Dictionary<PlayerPosition, int>
-flippedCardPositions: Dictionary<PlayerPosition, bool>
-startButton_Click(sender:object,e:RoutedEventArgs)
-initializeGame()
-initializeCardsDealtInPositions()
-initializeFlippedCardPositions()
-moveCard(cardImage:Image,absoluteX:int,
absoluteY:int)
-changeImage(cardImage:Image,card:Card)
-addToTextDisplay(textToAdd:string)
{local variable}
MoveDialogBox
-moveChosen: bool
-submitButton_Click(sender:object,e:RountedEventsArgs)
-Window_Closing(sender:object,e:CancelEventArgs)
PlayerFactory
+createHumanPlayer(moveChooser:IMoveChooser,
position:PlayerPosition,
stack:int): Player
+createAutoPlayer(position:PlayerPosition,
stack:int): Player
-createPlayer(moveChooser:IMoveChooser,position:PlayerPosition,
stack:int): Player
AutomatedMoveChoiceStrategy
<<Enumeration>>
PlayerPosition
+Right
+Left
+Middle
+Top
{parameter, returns}
Card
-cardIndex: int
-image: BitmapImage
-suit: string
-value: int
-determineSuit(index:int): string
-determineValue(index:int): int
+getImage(): BitMapImage
{parameter}
position
1
cards {List}
2
moveChooser
1
<<Interface>>
IMoveChoiceStrategy
+getMove(currentDealState:DealState,legalMoves:List<Move>,
cards:Card[]): Move
Player
+startingStack: int
+currentStack: int
+betInRound: int
+folded: bool
+makeMove(dealState:currentDealState): Move
+receiveCard(card:Card)
+reset()
{parameter}
moveChoice
1
<<Abstract>>
Move
+name: string
+amountPotRaised: int
+currentBetRaised: int
+amountIsNeeded(): bool
Fold
Call
Check
Raise
{return}
{parameter}
LegalMoveGenerator
+generateLegalMoves(currentDealState:DealState,
player:Player): List<Move>
{return}