Contract CO1: startGame

Operation:	startGame()
Cross References:	Use Cases: Start Game
Preconditions:	The system is set to create a game with two players – one sequential and one parallel.
Postconditions:	An instance of Game, *game*, was created
	An instance of GameStatistics, *gameStats*, was created
	Attributes of *gameStats* were set based on simulation

Contract CO2: setPreferences

Operation:	setPreferences(playerCount, playerTypes{List}, gameCount)
Cross References:	Use Cases: Set Preferences
Preconditions:	The application has initialized properly
	Parameter menus are functioning properly
	An instance of Game, *game*, exists
Postconditions:	*game.playerCount* was set to playerCount
	game.gameCount was set to gameCount
	player, an instance of HumanPlayer, was created
	Instances of other Players were created corresponding to each playerType

Contract CO3: raise

Operation:	raise(amount)
Cross References:	Use Cases: Take Turn
Preconditions:	The game is in play
	It is the user's turn in the game
	An instance of Game, *game*, exists
	player[i] is currentPlayer
	player[i] is an instance of HumanPlayer
	player[i+1] is nextPlayer
Postconditions:	*game.pot* was increased by amount
	player[i].stack was decreased by amount
	player[i]'s turn ended
	player[i+1]'s turn began

Contract CO4: check

Operation:	check()
Cross References:	Use Cases: Take Turn
Preconditions:	The game is in play
	players{list} are playing game
	player[i] is currentPlayer
	player[i] is an instance of HumanPlayer
	player[i+1] is nextPlayer
Postconditions:	*game.pot* was increased by the amount of the last raise
	player[i].stack was decreased by amount of the last raise
	player[i]'s turn ended
	player[i+1]'s turn began

Contract CO5: check

Operation:	call()
Cross References:	Use Cases: Take Turn
Preconditions:	The game is in play
	players{list} are playing game
	player[i] is currentPlayer
	player[i] is an instance of HumanPlayer
	player[i+1] is nextPlayer
Postconditions:	*player[i]*'s turn ended
	player[i+1]'s turn began

Contract CO6: fold

Operation:	fold()
Cross References:	Use Cases: Take Turn
Preconditions:	The game is in play
	players{list} are playing game
	player[i] is currentPlayer
	player[i] is an instance of HumanPlayer
	player[i+1] is nextPlayer
Postconditions:	*player[i]*'s turn ended
	player[i+1]'s turn began
	player[i] was removed from *players{list}*