

### Class Dealer

-mainDeck: MainDeck -discardDeck: Deck\* +Dealer() ~Dealer() makeDeck(): void resetDeck(): void deal(): Card\* +discard(card:Card\*): void +discard(deck:Deck\*): void +burn(): void +deleteAll(): void

# Class T2Poker

### Class Player

#playerNum: int #money: int #folded: bool #pocket: SmallDeck\* #pocket: SmallDeck\*
+Player()
+Player(num:int)
+-Player(num:int)
+-Player()
+getNum(): int
+inRound(): bool
+giveCard(card:Card\*): void
+getPocket(): SmallDeck\*
+fold(): SmallDeck\*
+getMoney(): int
+takePot(pot:int): void
+unFold(): void
+makeBet(comm:SmallDeck,minBet:int): Bet
+makeBig(minBet:int): Bet
+makeBig(minBet:int) +print(): void
+isHuman(): bool

Class Hand +cardArray: Card\* -player: Player\* -length: int -valid: int type: int tieBreaker: int +Hand(p:Player\*) +Hand() +~Hand() +putInCards(pocket:SmallDeck,community:SmallDeck): void +evaluateCards(): void +getPlayer(): Player\* +getType(): int -pair(): int -twoPair(): int -threeOfaKind(): int straight()
flush(): int fullHouse(): int fourOfaKind(): int

-quickSort(arr[]:Card,left:int,right:int): void

# Human Player Class

+bet(minimumBid:int,gameDisplay:displayD1&): int
+discard(gameDisplay:displayD1&): int
+ante(minimumAnte:int,gameDisplay:displayD1&): int
+HumanPlayer(displayPos:int,initialMoney:int)

# Al Class

+bet(minimumBid:int,gameDisplay:displayDl5): int
+discard(gameDisplay:displayDl5): int
+date(anteamount:int,gameDisplay:displayDl5): int
+AI(displayPos:int,startingMoney:int)

#### Class Game

-gui: Gui -dealer: Dealer\* -playerList: PlayerList\* -commCards: SmallDeck\* -winner: Player\* -minBet: int -centerPot: int -centerPot: int
-numFolded: int
-state: int
-userInputState: int
-currentPlayer: int
-handsUp: bool
-currentBet: Bet +Game() +~Game()

+startGame(): bool +playRound(): bool +endGame(): void +getWinner(): Player\* +getVille(): int +getGui(): Gui +getState(): int +getRoundNum(): int

+getButNum(): int
+getMinBet(): int getCurrentNum(): int

+getMinBet(): int
+getHandsUp(): bool
+getUrserNum(): int
+getHandsUp(): bool
+getUserInputState(): int
+getPlayerList(): PlayerList
+getCommCards(): SmallDeck
+getCurrentBet(): Bet
+setUserInputState(state:int): void
-processBet(Bet:bet, p:Player\*): void
-initialTurns(): void
-initialTurns(): void
-initialTurns(): void
-little(): void
-little(): void
-little(): void
-little(): void
-blinds(): void
-lop(): bool
-turn(): bool
-river(): bool
-river(): bool
-river(): bool
-ridgoundWinner:int): void
-discardAll(): void
-findBestHand(): Hand\*

# Class ComputerPlayer

+ComputerPlayer(num:int) +makeBet(C:SmallDeck,minBet:int): Bet calculateBet(): Bet

# Class T2HumanPlayer

+HumanPlayer(num:int) +makeBet(comm:SmallDick,minBet:int): Bet

# Class PlayerNode

data: Player\*
next: PlayerNode\*
prev: PlayerNode\* +PlayerNode(player:Player\* +~PlayerNode() +getPlayer(): Player\* +print(): void

#### Class Bet

-action: int -amount: int +Bet() +Bet(BetType:int,Amt:int) +print(): string +getAction(): int +getAmount(): int

# Class Layout

-size: Point
-center: Point
-deckArea: Point
-deckArea: Point
-domacrdArea: Point
-commCardArea: Point
-playerInfoArea: Point
-playerCardArea: Point
-playerCardArea: Point
-humanCards: SmallDeck
-compCards: SmallDeck
-layoutf()

+Layout() +~Layout() +setup(x:int,y:int): void +setBlankCards() +setup(x:int,y:int): void
+setlankCards()
+foldPlayer(pNum:int): void
+setPlayercards(): void
+setPlayercards(): void
+getSize(): Point
+getDeckArea(): Point
+getComCardArea(): Point
+getComCardArea(): Point
+getComCardArea(): Point
+getPlayerInfoArea(i:int): Point
+getPlayerInfoArea(i:int): Point
+getPlayerCardArea(i:int): Point
+getPlayerCardArea(i:int): SmallDeck
-setSize(x:int,y:int): void
-setDeckArea(): void
-setDeckArea(): void
-setCommCardArea(): void
-setCenterPotArea(): void
-setPlayerInfoArea(): void
-setPlayerInfoArea(): void
-setPlayerCardArea(): void

# Class PlaverList

head: PlaverNode\*

tail: PlayerNode\* button: PlayerNode\* current: PlayerNode +addPlayer(player:Player\*): void +PlayerList() +-Player() +getCurrentPlayer(): Player\* +next(): void +iterate(): void +purge(minBet:int): int +getButton(): Player +getSize(): int +print(): void +remove(): void +removeAll(): void -advanceButton(): void -reset(): void

# Class Point

-xCoord: int yCoord: int +Point()
+-Point()
+getX(): int
+getY(): int
+set(x:int,y:int): void
+print(): string

# Class Card

suit: int value: int next: Card\* prev: Card\* -Card() -Card(s:int,v:int) -Card() +getValue(): int +getSuit(): int print(): void isBlank(): void

# Class SmallDeck

-SmallDeck() ~SmallDeck() -getCard(num:int): Card -getSize(): int

# Class Deck

-top: Card\* -bottom: Card\* -size: int -cards[]: int -cardsLeft: int +Deck() +~Deck() +add(newCard:Card\*): void +add(\*d:Deck): void +draw(): Card\* +isEmpty(): bool +print(): void +deleteAllCards(): void +shuffle(): void -getNextCard(): int -isPair(a:int,b:int): bool

# Class Gui

+Gui()
+-Gui()
+-Gui()
+startGui(): bool
+showRoundWinner(roundWinner:int): void
+update(status:int): void
+getBetInput(minBet:int): Bet -delay(delay:int) -detectResize(sig :int): void -detectRes12e(sig :int): Void -redraw(): Void -clearScreen(): Void -displayCommon(): Void -displayPlayers(): Void -displayUserRegion(p:Player): Void -getBetType(minBet:int): int -aetGetAmount()

# Class MainDeck

+MainDeck() +~MainDeck() +shuffle(): void swap(source:Card\*.target:Card\*): void