Straights UML

startGameLoop() will call StraightModel's loopThroughPlayers() with handlePlayer passed in to dynamically handle both human and AI players with low coupling

StraightsController

+ startGameLoop()

- handlePlayer(HumanPlayer*)

- handlePlayer(ComputerPlayer*)

model

view

StraightsModel

+ dealHands();

+ loopThroughPlayers(Player& start, void (*func)(Player*))

+ whoHasCard(Card&): Player&;

+ isLegalPlay(Card&): bool;

View

+ *displayBoard()*

+ *displayHand(Card[]);*

+ *displayLegalPlays(Card[]);*

+ *displayMessage(string);*

+ *displayError(string);*

+ *promptCardSelection(): string*

+ *displayScore(discards Card[], int oldScore, int scoreGained)*

+ *displayWin(playerName: string)*

Extends

TextView

players

deck

Deck

+ shuffle();

+ isValidCard(): bool

+ stringToCard(): Card*

4

Player

+ name: int;

+ hasCard(Card&): bool

+ playCard(Card&);

+ discardCard(Card&)

+ giveCard(Card&);

+ calculateScore(): int

+ clearCards();

52
Card

1   Hand   0..13

Extends

Extends

HumanPlayer

ComputerPlayer

+ *doTurn();*

ComputerPlayer uses the Strategy design pattern to make employing different AI behaviour easy

1
turnStrat

TurnStrategy

+ doTurn();

Extends

SimpleStrategy