Straights UML (Final)

StraightsController
+ startGameLoop()

+ handlePlayer(HumanPlayer&)

+ handlePlayer(ComputerPlayer&)

Player Handler
setLoopFlag(bool): bool

+ shouldLoop(): bool

+ *handlePlayer(HumanPlayer&)*

+ *handlePlayer(ComputerPlayer&)*

PlayerHandler is the Visitor portion of the Visitor design pattern, implemented with the Player subclasses

Extends

model

view

Extends

StraightsModel
+ dealHands();

+ loopThroughPlayers(Player*, PlayerHandler&)

+ forEachPlayer(std::function<void(Player&)>)

+ addPlayer(std::unique_ptr<Player>)

+ replacePlayer(Player&, std::unique_ptr<Player>): Player*

+ whoHasCard(Card&): Player*;

+ getCard(string): Card*

+ playCard(Player&, Card&)

+ isLegalPlay(Card&): bool;

+ getLegalPlays(Player&): vector<Card*>

+ shuffleDeck()

+ isEndOfRound(): bool

+ isEndOfGame(): bool

+ getWinners(): vector<Player*>

+ resetRound()

+ printDeck()

View
+ *displayBoard()*

+ *displayHand(vector<Card*>);*

+ *displayLegalPlays(vector<Card*>);*

+ *displayMessage(string);*

+ *displayError(string);*

+ *promptCardSelection(): string*

+ *promptCommand(): string*

+ *displayScore(string playerName, vector<Card*> discards, int oldScore, int newScore)*

+ *displayWin(string playerName)*

TextView

players

4

deck

Player
+ hasCard(Card&): bool

+ removeCard(Card&)

+ discardCard(Card&)

+ giveCard(Card&);

+ calculateScore(): int

+ reset();

+ *accept(PlayerHandler&);*

Deck
+ DEFAULT_SEED: int

+ getCard(string rep): Card*

+ dealCard(Player&)

+ reset()

+ shuffle()

+ printDeck()

1

1

Discards

cards

```
                                          52
                        ┌──────────────────────┐
                  Hand  │         Card         │
                        ├──────────────────────┤                1  ┌─────────┐
                        │ + hasRank(Rank) bool  │◆──rank──▶│    Rank  │
                  0..13 │                      │              1   └─────────┘
                        │ + getStringRep(): string │
                        │                      │       1
                  0..13 │ + isFrontAdjacentTo(Card&) │◆
                        │                      │                1  ┌─────────┐
                        │ + isBackAdjacentTo(Card&) │      suit──▶│   Suit   │
                        └──────────────────────┘                  └─────────┘
```

Card

+ hasRank(Rank) bool

+ getStringRep(): string

+ isFrontAdjacentTo(Card&)

+ isBackAdjacentTo(Card&)

52

Hand

0..13

0..13

1

1

rank

Rank

1

1

suit

Suit

Extends

Extends

ComputerPlayer

+ *accept(PlayerHandler&)*

+ *doTurn();*

1

HumanPlayer

+accept(PlayerHandler&)

turnStrat

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

TurnStrategy

+ *doTurn();*

Extends

SimpleStrategy

+ doTurn();