Visual - rules:string + Init(player one, player two) << constructor>> + help() + click save button() + click load button() + surrender() + buttons card(player one, player two, card) + show game() + show winner() + show buttons cards() + show save button() + show labels buttons() + hide board() + first turn() + second turn() + resize cards() + config image played card(card) + config image(deck player one, deck player two, especial card) + get_player() + set player(player)

+ set rules(rules) + set game(game)

Game<abstract class> - player1:Player - player2:Player - rules:string - deck:Deck + init(deck, player one, player two) <base operation> + verify turn() + next round() + sum score() + reset values() + verify number() + verify suit(card) + draw() + save game() + load game() + get especial card() + get suit(card) + set deck(deck) + set especial card(especial card) Durak<controller> - rules:string - deck:Deck + init() override + play() <template method> + verify turn() override + next round() override + sum score() override + reset values() override + verify number() override + verify suit(card) override + draw() override + save game() override + load game() override + get special card() override + get suit(card) override + set deck(deck) override + set especial card(especial card) override

Deck - suits:vector<string> - values:vector<int> - deck:vector<string> + init() <<constructor>> + create deck() + deal cards() + get deck()

Player

- hand:vector<Card>
- value:int
- name:string
- playing card:string
- + append card(card)
- + remove card(card)
- + clean hand()
- + get_playing_card()
- + get_player_name()
- + get_size_hand()
- + get value()
- + get hand()
- + set hand()
- + set_value(value)
- + set player name(name)
- + set value hand(card)