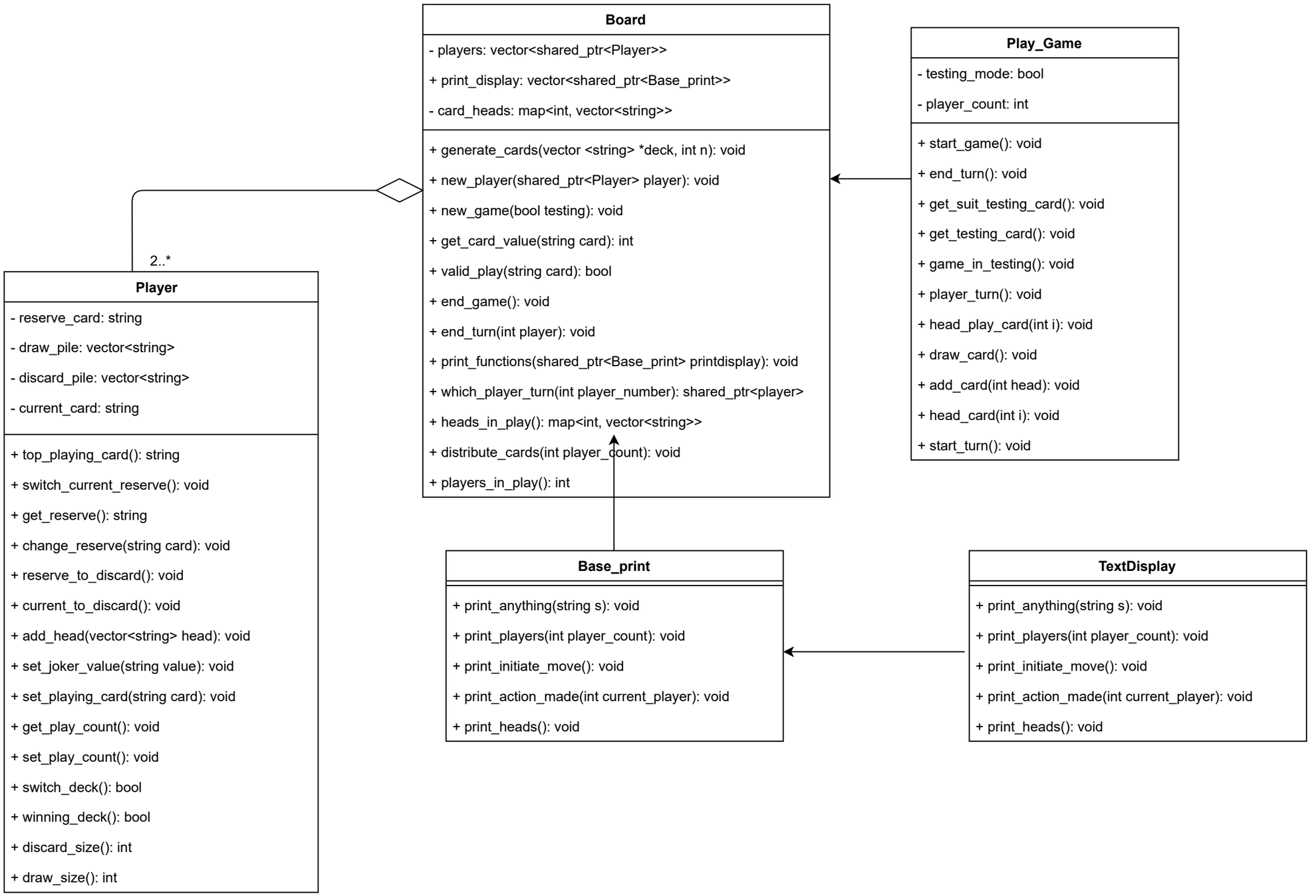
Board
- players: vector<shared_ptr<Player>>
+ print_display: vector<shared_ptr<Base_print>>
- card_heads: map<int, vector<string>>
+ generate_cards(vector <string> *deck, int n): void
+ new_player(shared_ptr<Player> player): void
+ new_game(bool testing): void
+ get_card_value(string card): int
+ valid_play(string card): bool
+ end_game(): void
+ end_turn(int player): void
+ print_functions(shared_ptr<Base_print> printdisplay): void
+ which_player_turn(int player_number): shared_ptr<player>
+ heads_in_play(): map<int, vector<string>>
+ distribute_cards(int player_count): void
+ players_in_play(): int
Play_Game
- testing_mode: bool
- player_count: int
+ start_game(): void
+ end_turn(): void
+ get_suit_testing_card(): void
+ get_testing_card(): void
+ game_in_testing(): void
+ player_turn(): void
+ head_play_card(int i): void
+ draw_card(): void
+ add_card(int head): void
+ head_card(int i): void
+ start_turn(): void
2..*
Player
- reserve_card: string
- draw_pile: vector<string>
- discard_pile: vector<string>
- current_card: string
+ top_playing_card(): string
+ switch_current_reserve(): void
+ get_reserve(): string
+ change_reserve(string card): void
+ reserve_to_discard(): void
+ current_to_discard(): void
+ add_head(vector<string> head): void
+ set_joker_value(string value): void
+ set_playing_card(string card): void
+ get_play_count(): void
+ set_play_count(): void
+ switch_deck(): bool
+ winning_deck(): bool
+ discard_size(): int
+ draw_size(): int
Base_print
+ print_anything(string s): void
+ print_players(int player_count): void
+ print_initiate_move(): void
+ print_action_made(int current_player): void
+ print_heads(): void
TextDisplay
+ print_anything(string s): void
+ print_players(int player_count): void
+ print_initiate_move(): void
+ print_action_made(int current_player): void
+ print_heads(): void