

Yasmine Siala

Professor Veenstra

CSE13S

4 November 2024

Assignment 3

letter_of_symbol()

1. Purpose: To return a specific character based on the value of Symbol sym
2. Parameters: sym Symbol
3. Return value: DOT returns '.', LEFT returns 'L', CENTER returns 'C', RIGHT returns 'R'
4. Pseudocode:
 - Check if sym == DOT, if true, return '.'
 - Check if sym == LEFT, if true, return 'L'
 - Check if sym == RIGHT, if true, return 'R'
 - Check if sym == CENTER, if true, return 'C'
 - If sym is none of these values return error

min()

1. Purpose: Compare two integer values and return the smaller one
2. Parameters: int a, int b
3. Return value: returns smaller int
4. Pseudocode:
 - Check if int a is <= int b and if true, return int a
 - If false, return int b

rand_roll()

1. Purpose: Use randi() to generate a random integer and then convert it to an integer 0-5
2. Parameters: void
3. Return value: int
4. Pseudocode:
 - Call randi() to generate a random int
 - Use %6 to make sure the number is between 0 - 5

left_of()

1. Purpose: Return the player number of the player to the left of 'player'.
2. Parameters: int player, int num_players
3. Return value: int
4. Pseudocode:
 - Check 'player' == num_players - 1: if true, return 0
 - Otherwise, return player + 1

right_of()

1. Purpose: Return the player number of the player seated to the right of the specified 'player'
2. Parameters: int player, int num_players
3. Return value: int
4. Pseudocode:
 - Check if 'player' == 0: if true, return num_players - 1
 - Otherwise, return player - 1

print_scores()

1. Purpose: To display the current scores of all players and the number of chips in the pot
2. Parameters: int num_players
3. Return value: void
4. Pseudocode:
 - Print header CURRENT_SCORES
 - Initialize sum_of_player_scores to 0 to track total chips
 - Using for loop, for each play (0 - num_players - 1):
 - Print player's score and name
 - Add player's score to sum_of_player_scores
 - Calculate chips in the pot using $3 * \text{num_players} - \text{sum_of_player_scores}$
 - Print the number of chips in the pot

play_game()

1. Purpose: Simulate the game by using a random seed, managing each player's turn, updating scores, and stopping when one player has all the chips
2. Parameters: int seed, int num_players
3. Return value: void
4. Pseudocode:
 - Initialize randi() with seed
 - Set each player's initial score in score[] to 3
 - Initialize pot to 0 to track the chips in the pot
 - While more than 1 player has chips:
 - For each player:
 - Check if player has chips, if they don't then skip to the next player

- Calculate how many players have chips and end game if only one player has chips
- Determine the number of dice to roll based on chip count
- For each die roll:
 - Use randi() to generate number 0-5
 - Based on the roll (if statement):
 - If DOT: do nothing
 - If LEFT: transfer one chip using left_of(), update score[], and decrement current player's score
 - If RIGHT: transfer one chip using right_of(), update score[], and decrement current player's score
 - If CENTER: add 1 chip to pot and decrement current player's score
 - Check that more than one player has chips left in score[], if not, game ends
- Print end message