Yahtzee game code:

import random

def roll\_dice(num\_dice):

"""Roll a specified number of dice."""

return [random.randint(1, 6) for \_ in range(num\_dice)]

def display\_dice(dice):

"""Display the current dice values."""

print("Dice:", ', '.join(str(d) for d in dice))

def choose\_dice():

"""Prompt the player to choose which dice to keep."""

keep = input("Enter dice indices to keep (e.g., 1 3 5), or 'r' to reroll all: ")

if keep.lower() == 'r':

return None

else:

return [int(i) - 1 for i in keep.split()]

def score(dice, category):

"""Calculate the score for a given category."""

if category == 'yahtzee' and len(set(dice)) == 1:

return 50

elif category == 'chance':

return sum(dice)

elif category in {'ones', 'twos', 'threes', 'fours', 'fives', 'sixes'}:

return sum(d for d in dice if d == int(category[-1]))

else:

return 0

def main():

print("Welcome to Yahtzee!")

categories = {'ones', 'twos', 'threes', 'fours', 'fives', 'sixes', 'chance', 'yahtzee'}

scorecard = {category: None for category in categories}

total\_score = 0

while categories:

print("\nRemaining categories:", ', '.join(categories))

category = input("Choose a category: ")

dice = roll\_dice(5)

rolls\_left = 2

while rolls\_left > 0:

display\_dice(dice)

keep = choose\_dice()

if keep is None:

dice = roll\_dice(5)

rolls\_left -= 1

else:

dice = [dice[i] for i in keep] + roll\_dice(5 - len(keep))

category\_score = score(dice, category)

scorecard[category] = category\_score

total\_score += category\_score

print(f"Scored {category\_score} points for {category}. Total score: {total\_score}")

del categories[category]

print("\nGame over! Final score:", total\_score)

if \_\_name\_\_ == "\_\_main\_\_":

main()

Simple testing strategy for the Yahtzee game:

1. Functionality Testing:

Dice Rolling:

- Verify that the `roll\_dice` function generates the correct number of dice values between 1

and 6.

Displaying Dice:

- Check if the `display\_dice` function correctly prints the dice values.

Choosing Dice:

- Test the `choose\_dice` function by entering different inputs (valid and invalid) to ensure it

handles them appropriately.

Scoring Categories:

- Manually calculate scores for various dice combinations and verify if the `score` function

returns the expected results.

2. Game Flow Testing:

Initialization:

- Start the game and verify that it displays the welcome message and available categories.

Category Selection:

- Select different categories and check if the game responds correctly, updating the scorecard

and total score accordingly.

Dice Rolling:

- Play through the game, ensuring that dice are rolled, displayed, and re-rolled according to the

player's choices.

Game Over:

- Play until all categories are filled and ensure the game ends with the correct final score.

3. Input Validation:

Invalid Inputs:

- Enter invalid inputs during category selection, dice choosing, and any other user prompts,

and ensure the game handles them gracefully.

Boundary Inputs:

- Test boundary cases, such as choosing to reroll all dice or keeping all dice, to ensure the

game functions correctly in these scenarios.

4. Category Scoring Testing:

Special Categories:

- Test special categories like Yahtzee and Chance with different dice combinations to ensure

they are scored correctly.

Number Categories:

- Verify that categories for numbers (ones, twos, threes, etc.) are scored accurately based on

the dice values.

5. Usability Testing:

User Experience:

- Evaluate the overall user experience, including interface clarity, ease of understanding, and

intuitiveness of game mechanics.

Error Handling:

- Assess how the game handles errors, provides feedback, and guides the player towards

correct actions.

6. Regression Testing:

- After making any changes to the code, repeat the above tests to ensure that existing

functionalities have not been affected.

7. Peer Testing:

- Have someone else play the game and provide feedback on their experience, any issues

encountered, and suggestions for improvement.

**Test Cases:**

**Test Case 1: Valid Category Selection**

- Input: Choose category: ones

- Expected Output: Dice: 3, 2, 1, 5, 4

Enter dice indices to keep (e.g., 1 3 5), or 'r' to reroll all: 1 3 5

Scored 1 point for ones. Total score: 1

**Test Case 2: Invalid Category Selection**

- Input: Choose category: invalid

- Expected Output: Remaining categories: ones, twos, threes, fours, fives, sixes, chance, yahtzee

Choose a category: Invalid category! Please choose from the remaining categories.

**Test Case 3: Reroll All Dice**

- Input: Choose category: chance

After rolling the dice:

Dice: 2, 4, 3, 6, 1

Enter dice indices to keep (e.g., 1 3 5), or 'r' to reroll all: r

- Expected Output: After rerolling:

Dice: 6, 4, 2, 1, 5

Enter dice indices to keep (e.g., 1 3 5), or 'r' to reroll all: 1 2 3

Scored 18 points for chance. Total score: 18

**Test Case 4: Yahtzee Score**

- Input: Choose category: yahtzee

After rolling the dice:

Dice: 3, 3, 3, 3, 3

- Expected Output: Scored 50 points for yahtzee. Total score: 50

**Test Case 5: Invalid Dice Selection**

- Input: Choose category: ones

After rolling the dice:

Dice: 2, 4, 6, 1, 3

Enter dice indices to keep (e.g., 1 3 5), or 'r' to reroll all: 1 2 3 4 5

- Expected Output: Invalid dice selection! Please choose valid indices or 'r' to reroll all.

**Test Case 6: Game Over**

- Input: Choose category: ones (repeat for all remaining categories)

- Expected Output:

Remaining categories: (none)

Game over! Final score: (total score)