```python
import random

# List of valid shapes that the player can guess from
SHAPES = ['circle', 'square', 'triangle', 'heart', 'rectangle', 'oval']

# Converts user's input to a valid shape name (handles first letter or full
word which is case-insensitive)
def normalize_input(user_input):
    user_input = user_input.lower()
    for shape in SHAPES:
        if user_input == shape or user_input == shape[0]:
            return shape
    return None

# Generates a random list of 4 shapes for the secret code
def generate_secret_code():
    return [random.choice(SHAPES) for _ in range(4)]

# Prompts the user to enter a valid guess and returns a list of 4 shapes
def get_user_guess():
    while True:
        guess_input = input("Enter your guess (4 shapes separated by
spaces, [e.g. circle s T RECTANGLE] ): ").split()

        if len(guess_input) == 1:
            command = guess_input[0].lower()
            if command == "exit":
                print("You chose to exit the game. Goodbye!")
                exit()
            elif command == "retry":
                print("Restarting the game from Attempt 1...\n")
                return "RETRY"

        if len(guess_input) != 4:
            print("Please enter exactly 4 shapes.")
            continue

        invalid_shapes = []
        guess = []

        for shape in guess_input:
            normalized = normalize_input(shape)
            if normalized:
                guess.append(normalized)
            else:
                invalid_shapes.append(shape)

        if invalid_shapes:
            for shape in invalid_shapes:
                print(f" '{shape}' is not a valid shape. Please use valid
shape names or initials.")
            continue  # Ask the user to try again

        return guess

# Compares the user's guess to the secret code and returns two counts
def evaluate_guess(secret, guess):
    correct_place = 0
    correct_shape = 0
    secret_copy = secret.copy()
    guess_copy = guess.copy()
    # Checks for correct shape in the correct position
    for i in range(4):
        if guess[i] == secret[i]:
            correct_place += 1
            secret_copy[i] = None
            guess_copy[i] = None

    # Checks for correct shape in the wrong position
    for i in range(4):
        if guess_copy[i] is not None and guess_copy[i] in secret_copy:
            correct_shape += 1
            index = secret_copy.index(guess_copy[i])
            secret_copy[index] = None

    return correct_place, correct_shape

# Prints a summary of the game result
def print_game_summary(secret_code, final_guess, attempts, won):
    print("\n" + "-"*50)
    print("               Game Result Summary")
    print("-"*50)
    print("Secret Code     :", ' '.join(secret_code))
    print("Final Guess     :", ' '.join(final_guess))
    print("Total Attempts  :", attempts)
    print("Game Status     :", "WON" if won else "LOST")
    print("-"*50)

# The loop and replay handling of the main game
def main():
    print("Welcome to Master Mind - Shape Edition!\n")
    print("Valid shapes are: circle, square, triangle, heart, rectangle,
oval")
    print("You can enter the full shape name or just the first letter
(e.g., 'c' for circle).\n")
    print("Choose difficulty (easy/hard): ", end='')

    # Reads difficulty input here instead of in choose_difficulty()
    while True:
        difficulty_input = input().lower()
        if difficulty_input == "easy":
            difficulty = "easy"
            max_attempts = None
            break
        elif difficulty_input == "hard":
            difficulty = "hard"
            max_attempts = 10
            break
        else:
            print("Invalid input. Please enter 'easy' or 'hard': ", end='')

    # Shows exit and retry instructions once difficulty is selected
    print("\nYou can type 'exit' anytime to quit the game.")
    print("You can type 'retry' to restart from the beginning.\n")

    secret_code = generate_secret_code()
    attempts = 0

    while True:
        print(f"\nAttempt {attempts + 1}")
        guess = get_user_guess()
        # If the user types "retry", reset attempts to 0 and generate a new
secret code
        if guess == "RETRY":
            attempts = 0
            secret_code = generate_secret_code()
            continue

        correct_place, correct_shape = evaluate_guess(secret_code, guess)
        print(f"Correct shape in the correct place: {correct_place}")
        print(f"Correct shape but in the wrong place: {correct_shape}")
        attempts += 1

        # 💡 Shows a hint every 5 valid attempts
        if attempts % 4 == 0:
            index = random.randint(0, 3)
            print(f"\n💡 Hint (randomly) : One of the correct positions
is: Position {index + 1} → {secret_code[index]}")

        if correct_place == 4:
            print("You cracked the code! ")
            print_game_summary(secret_code, guess, attempts, True)
            break

        if difficulty == "hard" and attempts == max_attempts:
            print("You've reached the maximum number of attempts.")
            print_game_summary(secret_code, guess, attempts, False)
            break

    # Asks if the player wants to play again
    while True:
        replay = input("\nDo you want to play again? (yes/no): ").lower()
        if replay == "yes":
            print("\nStarting a new game...\n")
            main()
            return
        elif replay == "no":
            print("Thank you for playing the game. See you soon! 👋")
            return
        else:
            print("Please enter 'yes' or 'no'.")

# Start the program
main()
```