In [ ]:

```
from random import sample

def secret_generator():
    """Generates a 4-digit secret code consisting of unique numbers."""
    return ''.join(map(str, sample(range(10), 4)))

def count_bulls(secret_code, user_guess):
    """Calculates the number of Bulls."""
    return sum(1 for i in range(len(secret_code)) if secret_code[i] == user_gues

def count_cows(secret_code, user_guess):
    """Calculates the number of Cows."""
    return sum(1 for i in range(len(secret_code)) if user_guess[i] in secret_cod

def check_bulls_and_cows(bulls, cows, secret_code):
    """Generates feedback based on Bulls and Cows counts."""
    if bulls == 4:
        return f"Grade: {bulls} bull{'s' if bulls > 1 else ''}.\nCongrats! The s

    grade = []
    if bulls > 0:
        grade.append(f"{bulls} bull{'s' if bulls > 1 else ''}")
    if cows > 0:
        grade.append(f"{cows} cow{'s' if cows > 1 else ''}")

    return f"Grade: {', '.join(grade) or 'None'}.", False

def start_game():
    """Main game loop."""
    attempts = 1
    secret_code = secret_generator()
    print("The secret code is prepared: ****.")

    is_secret_code_revealed = False
    while not is_secret_code_revealed:
        while True:
            user_guess = input(f"Turn {attempts}. Enter your 4-digit guess:\n")
            if user_guess.isdigit() and len(user_guess) == 4 and len(set(user_gu
                break
            print("Invalid input. Enter 4 unique digits.")

        bulls = count_bulls(secret_code, user_guess)
        cows = count_cows(secret_code, user_guess)
        feedback, is_secret_code_revealed = check_bulls_and_cows(bulls, cows, se
        print(feedback)
        attempts += 1

def main():
    start_game()

if __name__ == "__main__":
    main()
```

In [ ]: