Number Guessing Game Project

Project Overview

Build a fun number guessing game where the computer picks a random number and the player tries to guess it. This project will teach you about random numbers, user input validation, and game logic.

Learning Objectives

- Work with the `random` module
- Practice while loops and conditional logic
- Implement user input validation
- Create engaging user experience with feedback
- Track game statistics

Step-by-Step Implementation

Step 1: Basic Game Structure (10 minutes)

Create a new file called `guessing_game.py` in your coding-journey folder:

python

```python
import random

def number_guessing_game():
    """Main game function"""
    print("🎯 Welcome to the Number Guessing Game!")
    print("=" * 40)
    print("I'm thinking of a number between 1 and 100.")
    print("Can you guess what it is?")

    # Generate random number
    secret_number = random.randint(1, 100)
    attempts = 0
    max_attempts = 7

    while attempts < max_attempts:
        try:
            # Get user input
            guess = int(input(f"\nAttempt {attempts + 1}/{max_attempts} - Enter your guess: "))
            attempts += 1

            # Check the guess
            if guess == secret_number:
                print(f"🎉 Congratulations! You guessed it in {attempts} attempts!")
                print(f"The number was {secret_number}!")
                break
            elif guess < secret_number:
                print("📈 Too low! Try a higher number.")
            else:
                print("📉 Too high! Try a lower number.")

        except ValueError:
            print("❌ Please enter a valid number!")
            continue

    else:
        # This runs if the loop completes without breaking
        print(f"\n💔 Game over! You've used all {max_attempts} attempts.")
        print(f"The number was {secret_number}. Better luck next time!")

# Run the game
if __name__ == "__main__":
    number_guessing_game()
```

Step 2: Test the Basic Game (5 minutes)

1. **Save the file** as `guessing_game.py`
2. **Run it:** `python3 guessing_game.py`
3. **Test different scenarios:**
    - Guess correctly
    - Use all attempts
    - Try entering letters (should handle gracefully)
    - Try numbers outside the range

Step 3: Add Enhanced Features (15 minutes)

Let's make it more engaging! Replace your code with this enhanced version:

python

```
import random

def get_difficulty():
    """Let player choose difficulty level"""
    print("\nChoose your difficulty:")
    print("1. Easy (1-50, 10 attempts)")
    print("2. Medium (1-100, 7 attempts)")
    print("3. Hard (1-200, 5 attempts)")

    while True:
        try:
            choice = int(input("Enter 1, 2, or 3: "))
            if choice == 1:
                return 50, 10, "Easy"
            elif choice == 2:
                return 100, 7, "Medium"
            elif choice == 3:
                return 200, 5, "Hard"
            else:
                print("Please enter 1, 2, or 3!")
        except ValueError:
            print("Please enter a valid number!")

def give_hint(secret_number, attempts):
    """Give hints based on attempts"""
    if attempts == 3:
        if secret_number % 2 == 0:
            print("💡 Hint: The number is even!")
        else:
            print("💡 Hint: The number is odd!")
    elif attempts == 5:
        if secret_number % 10 == 0:
            print("💡 Hint: The number is divisible by 10!")
        elif secret_number % 5 == 0:
            print("💡 Hint: The number is divisible by 5!")

def number_guessing_game():
    """Main game function"""
    print("🎯 Welcome to the Number Guessing Game!")
    print("=" * 40)

    # Get difficulty
    max_num, max_attempts, difficulty = get_difficulty()
```

```
print(f"\n🎮 {difficulty} Mode Selected!")
print(f"I'm thinking of a number between 1 and {max_num}.")
print(f"You have {max_attempts} attempts to guess it!")

# Generate random number
secret_number = random.randint(1, max_num)
attempts = 0
guesses = []

while attempts < max_attempts:
    try:
        # Get user input
```