



뭔가 프로그래밍 관련 글을 쓰고는 싶은데, 마땅히 소재가 없어서 ChatGPT의 힘을 빌려보기로 했습니다.
사실 소재가 아예 없는 건 아닌데, 빠르게 쓰는 데는 무리가 있는지라...

h 간단한 파이썬 게임 하나 만들어 줘.

물론입니다! 어떤 종류의 게임을 원하시나요? 저는 몇 가지 예시를 드릴 수 있습니다:

- 숫자 맞추기 게임: 컴퓨터가 1부터 100 사이의 임의의 숫자를 선택하고, 사용자가 그 숫자를 맞추는 게임입니다.
- 단어 맞추기 게임: 컴퓨터가 사전에서 무작위로 단어를 선택하고, 사용자가 그 단어를 맞추는 게임입니다.
- 간단한 텍스트 기반 RPG: 사용자가 여러 선택지 중 하나를 선택하여 몬스터와 싸움을 벌이는 간단한 텍스트 기반의 RPG 게임입니다.

위의 예시 중 어떤 게임을 원하시나요? 또는 다른 아이디어가 있다면 알려주세요!

갑자기 숫자야구 게임이 끌려서 부탁해 봤습니다,

h Bulls and Cows, 숫자 야구 게임은 어때?

좋습니다! Bulls and Cows 게임을 만들어보겠습니다. 아래는 간단한 숫자 야구 게임의 파이썬 코드입니다:

```

import random

def generate_secret_number():
    digits = list(range(10))
    random.shuffle(digits)
    return digits[:3]

def get_user_input():
    while True:
        user_input = input("세 자리 숫자를 입력하세요: ")
        if len(user_input) == 3 and user_input.isdigit():
            return list(map(int, user_input))
        else:
            print("세 자리 숫자를 입력해야 합니다.")

def compare_numbers(secret, guess):
    bulls = 0
    cows = 0
    for i in range(3):
        if guess[i] == secret[i]:
            bulls += 1
        elif guess[i] in secret:
            cows += 1
    return bulls, cows

def play_game():
    secret_number = generate_secret_number()
    attempts = 0
    while True:
        guess = get_user_input()
        attempts += 1
        bulls, cows = compare_numbers(secret_number, guess)
        print(f"Bulls: {bulls}, Cows: {cows}")
        if bulls == 3:
            print(f"축하합니다! 비밀 숫자를 맞췄습니다. 시도 횟수: {attempts}")
            break

play_game()

```

기존에 있던 유명한 게임이라 그런지, 코드가 함수로 구분도 잘 되어 있고 예외 처리도 훌륭합니다.
친절하게 게임 규칙도 설명해 주네요.

이 코드는 0부터 9까지의 숫자 중에서 임의의 세 자리 숫자를 생성하고, 사용자로부터 세 자리 숫자를 입력받아 두 숫자를 비교합니다. 사용자의 입력과 생성된 비밀 숫자 간에 숫자와 위치가 일치하는 부분은 "Bulls"로, 숫자는 일치하지만 위치가 다른 부분은 "Cows"로 표시됩니다. 사용자가 비밀 숫자를 정확히 맞출 때까지 게임은 계속됩니다.

위 코드를 실행하면 숫자를 입력하는 프롬프트가 나타나며, "세 자리 숫자를 입력하세요:"라는 메시지가 표시됩니다. 사용자는 세 자리 숫자를 입력하고 Enter 키를 누르면 결과가 표시됩니다. 게임은 사용자가 비밀 숫자를 맞출 때까지 계속됩니다.

이렇게 숫자 야구 게임을 만들어보았습니다! 원하시는 대로 커스터마이징하거나 추가 기능을 구현할 수 있습니다. 즐겁게 게임을 즐기시기 바랍니다!

```
$ python bulls_and_cows.py
세 자리 숫자를 입력하세요: 123
Bulls: 0, Cows: 0
세 자리 숫자를 입력하세요: 456
Bulls: 1, Cows: 1
세 자리 숫자를 입력하세요: 457
Bulls: 0, Cows: 1
세 자리 숫자를 입력하세요: 584
Bulls: 0, Cows: 2
세 자리 숫자를 입력하세요: 846
Bulls: 3, Cows: 0
축하합니다! 비밀 숫자를 맞혔습니다. 시도 횟수: 5
```

재미있긴 한데, 굳이 따지자면 두 가지 아쉬움이 남았습니다.

'맞혔습니다'가 아니라 '맞았습니다'가 맥락상 적절하고요.

그리고 원래 Bulls and Cows(숫자야구) 게임은 4자리 숫자 맞추기가 기본 아니던가요?

말하고 나니 좀 억지스러운 것 같긴 하네요.

그래도 물어봤습니다.

h 해당 코드는 3자리 숫자를 맞추는 게임인데, 원래 4자리 숫자를 맞추지 않아?



죄송합니다. 제가 오해를 했습니다. "Bulls and Cows" 게임은 일반적으로 4자리 숫자를 사용하는 것이 맞습니다. 여기에 4자리 숫자 버전의 코드를 제공해 드리겠습니다:



```

import random

def generate_secret_number():
    digits = list(range(10))
    random.shuffle(digits)
    return digits[:4]

def get_user_input():
    while True:
        user_input = input("네 자리 숫자를 입력하세요: ")
        if len(user_input) == 4 and user_input.isdigit():
            return list(map(int, user_input))
        else:
            print("네 자리 숫자를 입력해야 합니다.")

def compare_numbers(secret, guess):
    bulls = 0
    cows = 0
    for i in range(4):
        if guess[i] == secret[i]:
            bulls += 1
        elif guess[i] in secret:
            cows += 1
    return bulls, cows

def play_game():
    secret_number = generate_secret_number()
    attempts = 0
    while True:
        guess = get_user_input()
        attempts += 1
        bulls, cows = compare_numbers(secret_number, guess)
        print(f"Bulls: {bulls}, Cows: {cows}")
        if bulls == 4:
            print(f"축하합니다! 비밀 숫자를 맞췄습니다. 시도 횟수: {attempts}")
            break

play_game()

```

이 코드는 0부터 9까지의 숫자 중에서 임의의 네 자리 숫자를 생성하고, 사용자로부터 네 자리 숫자를 입력받아 두 숫자를 비교합니다. 사용자의 입력과 생성된 비밀 숫자 간에 숫자와 위치가 일치하는 부분은 "Bulls"로, 숫자는 일치하지만 위치가 다른 부분은 "Cows"로 표시됩니다. 사용자가 비밀 숫자를 정확히 맞출 때까지 게임은 계속됩니다.

다시 한 번 사과드리며, 위 코드는 4자리 숫자를 맞추는 "Bulls and Cows" 게임을 구현한 것입니다. 이 코드를 실행하면 네 자리 숫자를 입력하는 프롬프트가 나타나며, "네 자리 숫자를 입력하세요:"라는 메시지가 표시됩니다. 사용자는 네 자리 숫자를 입력하고 Enter 키를 누르면 결과가 표시됩니다. 게임은 사용자가 비밀 숫자를 맞출 때까지 계속됩니다.

원하는 대로 커스터마이징하거나 추가 기능을 구현하여 게임을 즐기시길 바랍니다!

```

$ python bulls_and_cows.py
네 자리 숫자를 입력하세요: 1
네 자리 숫자를 입력해야 합니다.
네 자리 숫자를 입력하세요: 1234
Bulls: 1, Cows: 1
네 자리 숫자를 입력하세요: 5678
Bulls: 0, Cows: 2
네 자리 숫자를 입력하세요: 6534
Bulls: 1, Cows: 0
네 자리 숫자를 입력하세요: 1278
Bulls: 0, Cows: 3
네 자리 숫자를 입력하세요: 2187
Bulls: 2, Cows: 1
네 자리 숫자를 입력하세요: 2184
Bulls: 1, Cows: 1
네 자리 숫자를 입력하세요: 1587
Bulls: 1, Cows: 2
네 자리 숫자를 입력하세요: 2837
Bulls: 2, Cows: 1
네 자리 숫자를 입력하세요: 2157
Bulls: 2, Cows: 0
네 자리 숫자를 입력하세요: 2367
Bulls: 1, Cows: 1
네 자리 숫자를 입력하세요: 2447
Bulls: 1, Cows: 0
네 자리 숫자를 입력하세요: 4457
Bulls: 1, Cows: 0
네 자리 숫자를 입력하세요: 81377
네 자리 숫자를 입력해야 합니다.
네 자리 숫자를 입력하세요: 8137
Bulls: 4, Cows: 0
축하합니다! 비밀 숫자를 맞했습니다. 시도 횟수: 13

```

너무 생각 없이 했더니 시도 횟수가 급상승했습니다.
 사실 자릿수는 기존 코드에서 숫자 3을 4로만 바꿔주면 해결되는 문제였는데, 단순노동 시켰다고 ChatGPT가 제 머리에 무슨 짓을 했나 봅니다.



맨 처음 예시 들었던 것 중에서 '간단한 텍스트 기반 RPG'라는 게 뭔지 궁금하더라고요.
 마지막으로 물어봤습니다.

h 혹시 앞서 말한 간단한 텍스트 기반 RPG도 만들어줄 수 있어?



물론입니다! 간단한 텍스트 기반 RPG 게임을 만들어보겠습니다. 아래는 예시 코드입니다:




```

import random

class Player:
    def __init__(self, name, health, attack):
        self.name = name
        self.health = health
        self.attack = attack

class Enemy:
    def __init__(self, name, health, attack):
        self.name = name
        self.health = health
        self.attack = attack

def generate_enemy():
    enemies = [
        Enemy("고블린", 20, 5),
        Enemy("오크", 30, 10),
        Enemy("드래곤", 50, 15)
    ]
    return random.choice(enemies)

def print_stats(player, enemy):
    print(f"[플레이어] {player.name} - 체력: {player.health}, 공격력: {player.attack}")
    print(f"[적] {enemy.name} - 체력: {enemy.health}, 공격력: {enemy.attack}")

def perform_attack(attacker, defender):
    damage = attacker.attack
    defender.health -= damage
    print(f"{attacker.name}가 {defender.name}를 공격하여 {damage}의 피해를 입혔습니다.")
    if defender.health <= 0:
        print(f"{defender.name}를 처치하였습니다!")

def play_game():
    player_name = input("플레이어의 이름을 입력하세요: ")
    player = Player(player_name, 50, 10)
    enemy = generate_enemy()

    print("=== 게임을 시작합니다 ===")
    print_stats(player, enemy)

    while player.health > 0 and enemy.health > 0:
        print("\n어떤 행동을 하시겠습니까?")
        print("1. 공격")
        print("2. 도망")
        choice = input("번호를 입력하세요: ")

        if choice == "1":
            perform_attack(player, enemy)
            if enemy.health > 0:
                perform_attack(enemy, player)
        elif choice == "2":
            print("도망치셨습니다. 게임을 종료합니다.")
            break
        else:
            print("잘못된 입력입니다. 다시 시도하세요.")

        print_stats(player, enemy)

    print("=== 게임이 종료되었습니다 ===")

```

```
play_game()
```

위의 코드는 텍스트 기반 RPG 게임을 구현한 것입니다. 게임은 플레이어와 적이 서로 번갈아가며 공격하는 방식으로 진행됩니다. 플레이어와 적은 각각 이름, 체력(health), 공격력(attack)을 가지고 있습니다. 플레이어의 초기 체력은 50이며 공격력은 10입니다. 적은 ``generate_enemy()`` 함수를 통해 랜덤하게 생성됩니다.

게임은 사용자로부터 행동을 입력받아 처리합니다. 행동은 "1"을 입력하여 공격하거나 "2"를 입력하여 도망칠 수 있습니다. 행동 선택 이후에는 플레이어

무료 버전이라 그런 건지 중간에 생성이 끊겼습니다.

코드는 생각했던 것보다 단순했지만, 클래스와 함수 구분하는 거 보면 제가 배워야 할 수준입니다.

```
$ python bulls_and_cows.py
플레이어의 이름을 입력하세요: 밤팔이
=== 게임을 시작합니다 ===
[플레이어] 밤팔이 - 체력: 50, 공격력: 10
[적] 오크 - 체력: 30, 공격력: 10

어떤 행동을 하시겠습니까?
1. 공격
2. 도망
번호를 입력하세요: 1
밤팔이가 오크를 공격하여 10의 피해를 입혔습니다.
오크가 밤팔이를 공격하여 10의 피해를 입혔습니다.
[플레이어] 밤팔이 - 체력: 40, 공격력: 10
[적] 오크 - 체력: 20, 공격력: 10

어떤 행동을 하시겠습니까?
1. 공격
2. 도망
번호를 입력하세요: 1
밤팔이가 오크를 공격하여 10의 피해를 입혔습니다.
오크가 밤팔이를 공격하여 10의 피해를 입혔습니다.
[플레이어] 밤팔이 - 체력: 30, 공격력: 10
[적] 오크 - 체력: 10, 공격력: 10

어떤 행동을 하시겠습니까?
1. 공격
2. 도망
번호를 입력하세요: 1
밤팔이가 오크를 공격하여 10의 피해를 입혔습니다.
오크를 처치하였습니다!
[플레이어] 밤팔이 - 체력: 30, 공격력: 10
[적] 오크 - 체력: 0, 공격력: 10
=== 게임이 종료되었습니다 ===
```

이 코드로는 플레이어가 고블린, 오크는 잡아도 드래곤은 죽어도 못 잡습니다.

여지를 주기 위해 치명타 확률도 추가하고, 회복 기능과 체력 및 공격력 상승 이벤트 같은 것을 추가하면 훨씬 풍성한 게임이 되겠네요.

나도 적도 계속 성장할 수 있도록 적절히 밸런스도 맞추고, 성 등반 같은 콘셉트의 로그라이크로 만들어도 괜찮겠다 싶고요.

아이디어가 샘솟습니다.



개인적으로 이전까지 ICT 주요 키워드였던 메타버스, 블록체인, NFT, 웹 3.0 등은 썩 좋게 보지 않고 앉았습니다.

그런데 이상하게 생성형 AI는 자꾸 기대를 하게 되네요.

원리를 제대로 이해하고 활용할 수 있다면, 삶에 큰 변화를 가져다줄 수 있을 것 같다는 생각입니다.

여하튼 ChatGPT 덕분에 놀면서 시간 잘 때웠네요.
이제 할 일 하러 갑니다~

