

(A)

나는 게임 개발자이다. 온라인 게임 '리니지' 같은 게임을 개발하여 출시했고 온라인 홍보 중이야. 인스타그램에 홍보 게시물을 올리는데 카드 뉴스형태의 10장 짜리 이미지를 만들려고 해. 10장의 순서를 개략적으로 말할 게.

1번째: 표지

2번째: 개요전달(마지막까지 보면 어떤 이점이 있는지?)

3번째: 보관유도(중요하니 보관해서 여러번 보라고 어필)

4~9번째: 구체적인 내용

10번째: 프로필 페이지 유도, 다른 추천 콘텐츠 유도, 보관 유도

위의 순서와 형식대로 각 페이지에 내가 개발한 게임 관련 콘텐츠를 구체적으로 채워줘. 그리고 또한 각 페이지의 배경 이미지를 만들기 위한 미드저니용 프롬프트도 함께 생성하고 알겠지?



네, 게임 개발자로서 인스타그램을 통한 게임 홍보 계획을 세우셨군요! '리니지'와 같은 온라인 게임을 개발하셨다니 정말 멋집니다. 카드 뉴스 형태의 이미지를 만드는 데 있어서 구체적인 콘텐츠와 각 페이지에 맞는 배경 이미지를 생성하기 위한 미드저니용 프롬프트를 제안해 드리겠습니다. 게임의 특성, 목표, 그리고 가치를 반영하여 구성해 보겠습니다. 단, 실제 게임의 세부 사항은 알지 못하므로 일반적인 가이드라인을 제공할 것입니다.

1. 표지

1. 표지

- 콘텐츠: 게임 로고와 대표 이미지, “새로운 모험이 시작된다!”
- 미드저니 프롬프트: “고대 판타지 세계의 대문이 열리는 순간을 포착한, 신비로운 색감과 함께 게임 로고가 돋보이는 아트웍”

2. 개요 전달

- 콘텐츠: 게임의 핵심 요소와 특징 간략 소개, “마지막까지 보면 어떤 이점이 있는지 알려드립니다!”
- 미드저니 프롬프트: “판타지 세계 지도를 배경으로, 게임의 핵심 요소와 특징이 간략히 소개되는 이미지”

3. 보관 유도

- 콘텐츠: “이 정보는 중요합니다! 보관해서 여러 번 보세요!”
- 미드저니 프롬프트: “사용자의 관심을 끌 수 있는 아이콘과 함께 ‘보관하세요’ 메시지가 강조된 이미지”

4~9. 구체적인 내용

4. 게임 세계의 설정

- 콘텐츠: “판타지 세계의 깊은 역사와 전설”
- 미드저니 프롬프트: “고대 서사시를 연상시키는, 판타지 세계의 역사와 전설이 담긴 책의 페이지”

5. 주요 캐릭터 소개

- 콘텐츠: “당신을 도와줄 전설 속 캐릭터들”
- 미드저니 프롬프트: “게임 내 주요 캐릭터들이 포즈를 취하며 서 있는 영웅적이고 가련

del B: gpt-4-0125-preview

6. 게임 플레이 메커니즘

- 콘텐츠: “모험, 전투, 그리고 동맹”
- 미드저니 프롬프트: “다이나믹한 전투 장면, 플레이어 간의 협력을 강조하는 이미지”

7. 특별 이벤트 및 보상

- 콘텐츠: “특별 이벤트 참여로 받을 수 있는 독점 보상”
- 미드저니 프롬프트: “보물 상자와 고유 아이템이 빛나는, 보상의 가치를 강조하는 이미지”

8. 커뮤니티 및 소셜 기능

- 콘텐츠: “함께하는 즐거움, 우리의 커뮤니티에 참여하세요”
- 미드저니 프롬프트: “플레이어들이 모여 있는 장소, 활발한 커뮤니케이션과 교류의 모습”

9. 개발자 메시지

- 콘텐츠: “게임을 만든 우리의 이야기”
- 미드저니 프롬프트: “개발자 팀이 모여 게임에 대해 논의하는 친근하고 진정성 있는 이미지”

10. 프로필 페이지 유도 및 추천 콘텐츠

- 콘텐츠: “더 많은 모험을 위해 프로필 페이지를 방문하세요. 다른 추천 콘텐츠도 확인해보세요!”
- 미드저니 프롬프트: “소셜 미디어 프로필 링크와 함께, 게임 내 다양한 활동을 제안하는

나는 게임 개발자이다. 아래 와 같은 게임을 개발하여 출시했고 온라인 홍보 중이야.

content = '고대 문명이 깃든 광대한 땅, 프리즘의 힘이 흩어진 곳입니다. 인간, 엘프, 드워프, 오크, 드래곤이 지배권을 놓고 경쟁합니다.'

스토리: 프리즘의 힘으로 고대 아틀란티스 문명이 멸망했습니다. 이제 프리즘 조각들은 흩어져 있고, 다양한 파벌들이 그 조각들을 차지하려고 합니다. 플레이어는 그 조각들을 모아서, 부패한 보스들을 물리치고, 프리즘의 멸망 뒤에 숨겨진 진실을 밝혀내야 합니다.

특징:

- Haven City, Elven Forest, Dwarven Mountains 등 다양한 지역 탐험
- 다양한 클래스 중에서 선택할 수 있으며, 각각 고유한 능력과 플레이 스타일을 갖추고 있습니다
- NPC와의 관계를 발전시키고 퀘스트를 완료하며 보상을 획득합니다
- 스킬, 콤보, 전략을 활용한 실시간 액션 전투로 전투
- 혼자 또는 친구들과 파티 놀이

게임 플레이 메커니즘:

- 레벨링, 스킬트리, 장비를 통한 캐릭터 개발
- 무기, 갑옷, 물약, 재료를 갖춘 아이템 시스템
- 금화, 은화, 동화를 사용한 화폐제도
- 퀘스트 관리, 인벤토리 및 캐릭터 정보를 위한 직관적인 UI

전투:

1번째 페이지: 표지

- 게임 제목: "Prism of Atlantis"
- 태그라인: "Unveil the truth behind the fallen civilization"
- 배경 이미지 프롬프트: A grand, ancient cityscape with a massive, glowing prism at the center. The city is in ruins, with hints of various races (human, elf, dwarf, orc, dragon) in the architecture.

2번째 페이지: 개요전달

- 고대 문명의 비밀과 힘을 찾아 떠나는 모험
- 다양한 지역 탐험과 선택한 클래스의 고유한 능력으로 전투
- NPC와의 관계 개발, 퀘스트 완료로 보상 획득
- 레벨링, 스킬트리, 장비를 통한 캐릭터 개발
- 혼자 또는 친구들과 파티 플레이
- 배경 이미지 프롬프트: A beautiful, full map of the game's world, showcasing the various regions like Haven City, Elven Forest, Dwarven Mountains, etc.

3번째 페이지: 보관유도

- 이 콘텐츠를 보관해 여러번 확인하세요.
- 게임의 다양한 특징과 메커니즘을 숙지하면 더 즐거운 플레이가 가능해요.
- 배경 이미지 프롬프트: A lush, vibrant image of an adventurer's gear, including a map, a compass, and some magical artifacts, spread out on a wooden table.

4번째 페이지: 구체적인 내용 - 스토리

- 프리즘 조각들이 흩어져, 다양한 파벌들이 경쟁
- 플레이어는 프리즘 조각들을 모아 부패한 보스들을 물리치고, 진실을 밝혀내야 함
- 배경 이미지 프롬프트: A dramatic, dynamic illustration of the moment the ancient Atlantean civilization was destroyed by the power of the prism, with the prism shattering into pieces.

5번째 페이지: 구체적인 내용 - 클래스

- 다양한 클래스 중 선택: 워리어, 메이지, 레인저, 그림, 신궁
- 각 클래스의 고유한 능력과 플레이 스타일
- 배경 이미지 프롬프트: A set of five character illustrations, each one showcasing a different class (warrior, mage, ranger, grim, and shinobi) in an action pose, with their unique abilities visualized.

6번째 페이지: 구체적인 내용 - 전투

- 실시간 액션 전투: 기본 공격, 스킬, 콤보
- 부패한 상사와 부하들을 물리치며 이야기 진행
- 전략과 팀워크를 활용한 도전
- 배경 이미지 프롬프트: An epic, dynamic battle scene illustration featuring a party of adventurers (each with a different class) fighting against a corrupted boss and its minions.

7번째 페이지: 구체적인 내용 - 아이템 시스템

- 다양한 아이템: 무기, 갑옷, 물약, 재료
- 아이템을 통한 캐릭터 개발과 강화
- 배경 이미지 프롬프트: A rich, detailed image of a bustling marketplace in one of the game's regions, with various items like weapons, armor, potions, and materials on display.

8번째 페이지: 구체적인 내용 - 화폐제도

- 화폐: 금화, 은화, 동화
- 화폐를 통한 거래, 아이템 구매, 서비스 이용
- 배경 이미지 프롬프트: An illustration of a treasure chest overflowing with gold, silver, and copper coins, with a few magical, glowing gems mixed in.

9번째 페이지: 구체적인 내용 - UI

- 직관적인, 유저 친화적인 UI
- 인벤토리, 캐릭터 정보, 퀘스트 관리
- 배경 이미지 프롬프트: A clean, well-organized screenshot of the game's UI, showcasing the inventory, character info, and quest management elements.

10번째 페이지: 프롱필 페이지 유도, 다른 추천 콘텐츠 유도, 보관 유도

- 프롱필 페이지: 자세한 정보와 게임 다운로드 링크 확인
- 추천 콘텐츠: 게임 플레이 영상, 팁과 트릭, 커뮤니티 활동 등
- 이 콘텐츠를 다시 한 번 보관해 주세요.
- 배경 이미지 프롬프트: A collage of images related to the game and its community, including gameplay screenshots, video thumbnails, and social media posts.

CodeChef 문제 중 난이도 최상급 문제 1개를 추천하고 그 내용을 한글로 설명해주세요.
단, 정규식 관련 문제는 제외하고요:
문제: -||-

당신은 세계 최고의 파이썬 개발자입니다. 위의 <문제>를 해결하는 파이썬 코드를 작성하세요.

위의 <문제>를 해결하는 로직을 PlantUML을 사용하여 Activity Diagram으로 작성해 주세요.

Diagram은 시작점, 조건 분기, 반복 구조를 명확하게 표현해야 합니다. 특히, PlantUML에서 사용되는

if 조건문의 올바른 문법과 구조를 사용하여, 조건에 따른 흐름을 정확하게 다루어야 합니다.

조건문을 사용할 때는 if (condition) then (outcome) 형태로 작성하고, 필요에 따라 else (alternative)를 추가해 주세요. 조건문의 끝에는 반드시 endif를 사용해야 합니다.

또한, 반복문을 사용할 때는 foreach 대신 while 구문을 적절히 활용하여 논리적인 흐름을 설명해야 합니다.

이 Diagram에서는 재귀 호출의 구조를 group 블록을 사용하여 내부 로직을 그룹화하고 설명하십시오,

[주] Diagram에서 foreach 대신 while 구문을 사용하여 같은 로직을 반복 구조로 표현해 주십시오.

PLANTUML용 diagram을 작성할 때 다음 사항을 잘 확인하고 반영해 주세요.

1. 명확한 제어 흐름 요청: '각 조건 분기마다 endif, end while 등을 명시해 주세요. 모든

추천드릴 난이도 최상급 문제로 "Sereja and Brackets" (문제 코드: SEABRCKT)를 선택하겠습니다. 이 문제는 괄호 문자열에서 부분문자열 중 가장 큰 길이의 균형잡힌 괄호 문자열을 찾는 문제입니다.

문제 설명

문제 개요:

- 주어진 괄호 문자열에서 가장 긴 균형 잡힌 부분 문자열의 길이를 찾는 문제입니다.

입력:

- 첫 번째 줄에 괄호 문자열 (s)이 주어집니다. 문자열 (s)의 길이는 최대 10^5 입니다.

출력:

- 첫 번째 줄에 가장 긴 균형 잡힌 괄호 문자열의 길이를 출력합니다.

예제 입력:

```
(()((())()))
```



예제 출력:

```
10
```



해결 방법

이 문제의 해결을 위해 구간 트리(Segment Tree)를 사용하여 각 구간에서 균형 잡힌 괄호 문자

해결 방법

이 문제의 해결을 위해 구간 트리(Segment Tree)를 사용하여 각 구간에서 균형 잡힌 괄호 문자열을 빠르게 찾을 수 있습니다. 각 노드에 대해 다음 정보를 저장합니다.

- 왼쪽에 남아있는 여는 괄호 '('의 수.
- 오른쪽에 남아있는 닫는 괄호 ')'의 수.
- 해당 구간 내에서 완전히 균형 잡힌 괄호 문자열의 길이.

파이썬 코드

```
class SegmentTree:
    def __init__(self, data):
        n = len(data)
        self.n = n
        self.tree = [None] * (4 * n)
        self.build(data, 0, 0, n - 1)

    def build(self, data, node, start, end):
        if start == end:
            if data[start] == '(':
                self.tree[node] = (1, 0, 0)
            else:
                self.tree[node] = (0, 1, 0)
        else:
            mid = (start + end) // 2
            left_child = 2 * node + 1
            right_child = 2 * node + 2
            self.build(data, left_child, start, mid)
```

```

def combine(self, left, right):
    open_left, close_left, match_left = left
    open_right, close_right, match_right = right

    match = min(open_left, close_right)
    return (open_left + open_right - match, close_left + close_right - match, match)

def query(self, l, r, node, start, end):
    if l > end or r < start:
        return (0, 0, 0)
    if l <= start and end <= r:
        return self.tree[node]

    mid = (start + end) // 2
    left_child = 2 * node + 1
    right_child = 2 * node + 2

    left_result = self.query(l, r, left_child, start, mid)
    right_result = self.query(l, r, right_child, mid + 1, end)

    return self.combine(left_result, right_result)

def get_longest_balanced(self, l, r):
    return self.query(l, r, 0, 0, self.n - 1)[2]

if __name__ == "__main__":
    s = input().strip()
    st = SegmentTree(s)

```

Activity Diagram

다음은 `Sereja and Brackets` 문제를 해결하기 위한 로직을 Activity Diagram으로 작성한 것입니다.

```

@startuml
start
:Input string s;
:Build segment tree for s;
while (queries exist?) is (true)
    :Read query range (l, r);
    :Calculate longest balanced substring using segment tree;
    :Output the result;
endwhile (false)
stop
@enduml

```

상세한 설명

- **Segment Tree 구축 및 쿼리:**
 - 각 노드는 `(open_count, close_count, balanced_length)` 정보를 저장합니다.
 - `combine()` 함수는 두 노드의 정보를 합쳐 새로운 노드의 정보를 생성합니다.
- **쿼리 처리:**
 - 주어진 구간에 대해 균형 잡힌 문자열의 길이를 빠르게 계산하기 위해 쿼리를

