|  |  |  |
| --- | --- | --- |
| 19.10.25 | 회의 일지 | 김기태 / 이경섭 |
|  | 기획회의3 | 작성자 : 이경섭 |
| 회의 내용 | 게임의 로직에 관하여 이야기 해보자.  단어를 조합해서 마법을 만든다.  단어는 최대 세단어까지 조합  수정 : 단어는 내부적으로 ‘형상’과 ‘속성’(소재)로 나눈다. 게임 플레이 안에서는 단어를 뜻하는 아이콘만 보여준다.(ex : 형상은 무색, 속성은 유색으로 아이콘을 만들어서 힌트 제공)  마법 동작 Flow  단어를 커맨드큐에 넣는다.(최대 3-4개)  비트값 검사  Ex) (0000 0000) (0000 0000) (0000 0000)  1번단어 2번단어 3번단어 -> 8비트의 멤버에 담는다고 할때, 최대 4개단어를 조합할 경우 32비트일것임.  8비트의 각 비트에 단어 하나를 할당해둔다. 8개 이상의 단어를 만들거라면 담아둘 변수의 크기를늘려야됨.  마법 실행시 검사를 통해 마법 객체를 완성한다.  Ex) (0000 0000) (0000 0000) (0000 0000) (초기상태)  1. 불(0000 0001) 입력(값은 임시)  1.1 변수 크기만큼 비트시프트 -> (0000 0000) (0000 0000) (0000 0000)  1.2 방금 입력한 단어를 or연산 (0000 0000) (0000 0000) (0000 0001)  2. 물(0000 0010) 입력  2.1 변수 크기만큼 비트시프트 -> (0000 0000) (0000 0001) (0000 0000)  2.2 방금 입력한 단어를 or연산 (0000 0000) (0000 0001) (0000 0010)  이를 반복한다.  그다음 case문으로 미리 Define 해둔 값을 지정해 마법 생성  Ex) #define 불+물 (0000 0000) (0000 0001) (0000 0010)  Case 불+물:  생성 불+물마법  Break;  Default:  폭!발!(실패)  장점 : 우리가 안만든 모든 케이스에 대해서는 실패판정 가능. 단 폭!발!이라는 형태로 고정됨.  단점 : 템플릿에 찍어내는거마냥 한번에 색깔바꿔가며 쭈르륵 뽑아내는 것같은건 어려움. 마법 하나당 하나의 case문과 define을 만들어야할 것.  그렇다면 마법의 개수는 어떻게 처리해야 될까? 상호작용의 개수는? | |
| 요약 | 핵심 기능인 마법 동장 que에관하여 이야기함 | |
| 향후 계획 | 향후 마법의 개수를 어떻게 할 것인지 시험 기간 끝나고 논의 | |