게임공학과 20학번 이규원

게임기획 1차 수정 3주 주간보고

캐릭터 별 스킬 상세 설정

스테이지 구성

맵 구성, 맵 크기

캐릭터 크기 설정

상세 내용은 기획서에 작성

데이터베이스

DBMS의 기능

* 데이터 중복의 최소화
* 동시성 제어
* 보안기능
* 다양한 사용자 인터페이스
* 데이터 사이에 존재하는 복잡한 관련성을 표현
* 데이터베이스의 무결성을 보장
* 백업과 복구 기능

DBMS 사용효과

* 표준화된 데이터 관리
  + 데이터의 중복이 없어짐
* 데이터 구조 변경에 융통성 부여
  + 데이터베이스 내의 자료 구조가 변경되어도 사용자에 대한 영향은 거의 없음
* 응용 프로그램의 개발 시간 단축
  + 응용프로그램의 상당한 부분을 DBMS가
* 최신의 정보 제공
  + 사용자 중에서 한사람의 갱신으로 나머지 사람은 바로 갱신된 값 접근 가능
* 규모의 경제성
  + 부서마다 다른 방식으로 자료를 관리 저비용, 관리할 데이터가 많을수록

DBMS 단점

* 비용 증가
  + 소잡는칼로 닭잡으면 손해임
* 성능 감소
  + 여러 사용자가 이용하지 않으면 필요없는 기능을 위해 구동될 수 있다
* 고장의 영향 확대
  + 한군대가 망가지면 다 망가질 수 있다.

데이터 화일의 구조

? : 배열과 연결리스트의 구조의 한계를 어떻게 벗어날까?

-> 배열들을 연결리스트로 (덱 구조), 더 나아가 트리구조로?

* 구조의 단어
  + 블록 : 배열
  + 레코드 : ex) 학생번호, 이름, 번호 등
* 배열/연결의 문제가 해결되었는가?
  + 크기가 고정? : 완벽x 어느정도 해소는 됨.(전체적으로 연결리스트 구조니까, 적게 잡았으면 새로운 블록 붙이면 된다.
  + 중간에 첨삭? : 배열의 크기가 크지 않아서 첨삭이 비교적 자유롭다
  + 중간의 노드를 액세스 할 때 오래걸림(하드디스크 액세스 횟수 문제) : 블럭마다 순차검색을 하니 비교적 완화된 것 같다.
* 블럭의 크기가 어느정도가 적당한가?
  + 관점 1 : 한블럭 안에 레코드를 몇 개 넣는게 좋을까?
  + 관점 2 : 블럭의 크기를 고정시킨다면? (그럼 레코드의 크기에 따라 블럭이 들어갈 개수가 달라질 수 있다
    - 레코드의 크기 -> OS에 따라 달라질 수 있음
    - 블럭의 크기 -> 설계자에 따라 가능.