게임공학과 20학번 이규원

공부내용 2주 주간보고

3D게임 프로그래밍

루트 시그너쳐

* 정의 : 그래픽스 파이프라인에서 셰이더와 애플리케이션 간의 리소스를 정의하는 구조

리소스뷰

* CBV(Constant Buffer View : 상수버퍼뷰) : 행렬, 조명 파라미터 등 자주 변하지 않지만 셰이더에서 반복적으로 사용하는 데이터를 저장
* SRV(Shader Resource View : 셰이더 리소스 뷰) : 셰이더에서 읽기 전용으로 사용할 수 있는 리소스를 참조하는 뷰
* UAV(Unordered Access View : 비순차적 접근 뷰) : 읽기와 쓰기가 모두 가능한 리소스, 여러 스레드가 동시에 해당 리소스에 접근할 수 있다.

텍스처 필터링

* 정의 : 하나 이상의 텍셀을 읽고 결합하여 하나의 색상을 생성하는 과정
* UV 좌표로 텍스처의 텍셀을 샘플링하는 방법에 영향을 줌
* 점 필터링 : 텍스처를 확대하거나 축소할 때 가장 가까운 텍셀만을 사용해 색상을 결정
* 선형 필터링 : 텍스처의 색상을 결정할 때 가장 가까운 4개의 텍셀의 색상 값을 선형 보간하여 중간 값을 계산
* 밉맵 필터링 : 여러 해상도의 텍스처를 사용해 축소시 품질을 유지

데이터베이스

파일 시스템

* 특징 : 파일을 이용하여 자료를 관리하는 방법
* 문제 : 데이터에 대한 프로그램의 의존도가 높다
  + 데이터의 중복성

Ex) 학생파일과 학과파일의 중복된 데이터가 존재하며 데이터가 서로 불일치하면(게임공학과, 멀티미디어 공학과) 어떤 데이터가 정확한 데이터인지 판단할 수 없어 치명적인 문제가 될 수 있다.

* 부족한 기능
* 문제 : 데이터에 대한 프로그램의 의존도가 높다

성능

* 시간복잡도 O(n)
* 공간복잡도 : 메모리(자료구조), 하드디스크(파일처리론)

하드디스크의 저장공간을 배열로 저장할 때 문제점

* 배열의 연결성을 통해 하드디스크의 어디에 저장되어 있는지 중요한 특성상 연결되어 있으면 빠르게 순차검색이 가능할 수 있다.
* 삽입/제거하면 앞으로 당겨야 하는데, 하드디스크에 있는 것들을 다 당기려면 액세스 하느라 연산이 너무 커진다.

연결리스트를 사용하면?

* 장점 : 추가, 제거가 배열에 비해 빠르고 뒤로 밀거나 하는 일들은 없다.
* 단점 : 탐색이 디스크를 뒤지면서 가야 하니 문제고, 주소를 저장할 공간적 디메리트도 있다, 순차 검색하면 거의 모든 디스크를 읽어야 할 수도 있다.

게임소프트웨어공학

소프트웨어 품질의 특징

* 사용자가 원하는 품질 수준이 점점 늘어나는데, 개발기간은 한정적이다.
* 소프트웨어, 하드웨어 기술이 급진적으로 발전한다.
* 고려해야할 게 엄청 많은데 눈에 안보인다.
* 소프트에어 품질은 개발자의 능력에 좌지우지 된다.
* 이해관계자가 수락할 수 있는 수준에 존재해야 한다.

소프트웨어 품질의 정의

* 명시적인 기능 및 성능 요구사항, 명시적으로 문서화된 개발 표준, 개발된 소프트웨어에서 기대되는 묵시적인 특성에 대한 적합성

소프트웨어 품질의 중요성

* 겉으로 보이는건 꽤 적다.
* 감춰진 증상이 더 많음
  + 변수명령, 보안코딩등의 규칙
* 사용자가 만족하는 품질의 소프트웨어(사용자가 합의한 수준의)
  + 정해진 시간 안에 정해진 비용 내에 소프트웨어 기능을 만든다..

소프트웨어 이해관계자

이해관계자별 기대치

스폰서 : 적응 비용으로 소프트웨어 개발

사용자 : 적은 비용으로 쉽게 즐기고싶다.

유지보수자 : 이해하기 쉬운 소스코드 등..

소프트웨어 외적 품질요소

* 정확성 : 원하는 결과가 생성하는지 판다
* 신뢰성 : 운영중에 오류의 발생정도
* 견고성 : 사용자가 제시하지 않은 환경에서도 소프트웨어가 합리적으로 동작
* 성능 : 소프트웨어의 효율성 ex) 알고리즘의 O(n)
* 사용자 친숙성 : 사용자가 이용하기 편리한지의 정도
* 가용성 : 사용가능한 시간 / 운영 시간 이 클수록 가용성이 높음
* 보안성 : 공격에도 올바르게 동작 // 취약점, 사고 통계, 보안으로 인한 손실액이 척도

유지보수성

* 수정 : 오류가 발생한 경우 활동
* 적응 : 소프트웨어 운영 조건에 대한 변화를 수용하는 유지보수 활동
* 완전 : 사용 중인 소프트웨어의 가독성, 이해성을 높이는 재구조화를 목적으로 수행되는 활동
* 예방 : 직렬로 두면 죽으니까 병렬로

프로세스 품질

* 프로세스 모델 적합성 : 어떤 모델?
* 개발 방법론 적합성 : 객체지향 방법, 구조적 방법론
* 도구 적합성 : 소프트웨어 개발 과정에서 지원 도구가 개발 환경에 적합한지
* 표준 준수성 : 선택한 표준이 적절하고, 선택한 표준을 준수해 프로젝트가 진행되고 있는지
* 프로젝트 데이터 관리 수준 : 데이터가 정보 저장소에 저장되어 신규 프로젝트의 예측 활동에서 활용할 수 있는지