게임 서버 프로그래밍 책 설계

지산

February 4, 2024

Contents

I 글쓰기 **7**

1 어떤 글을 써야 하나? **9**

1.1 쉬운 글 9

1.1.1 한자어를 피한다 9

1.1.2 짧게 쓴다 9

1.1.3 핵심을 쓴다 10

1.2 명확한 글 10

1.2.1 코딩하듯이 10

1.2.2 대상을 명확하게 안다 10

1.3 재미있어야 한다 11

1.4 반복하여 개선한다 11

1.5 함께 한다 11

2 예쁜 문서 만들기 **13**

2.1 라텍 잘 다루기 13

2.2 폰트 13

2.3 배치 13

3 LaTeX **15**

3.1 빈 페이지 15

3.2 한글 폰트 지정 15

3.3 프로젝트 구조 15

II 내용 **17**

4 무엇을 쓸 것인가? **19**

4.1 안내를 한다 19

4.2 정리를 한다 19

4.3 실제 내용들 19

III 라이브러리 **21**

5 연습 준비 **23**

6 라이브러리 목록 **25**

6.1 표준 라이브러리 25

6.2 boost와 신규 라이브러리 26

6.3 tbb 26

6.4 메모리 관리 26

6.5 통신 26

6.6 데이터 관리 27

6.7 공간과 물리 처리 27

6.8 메트릭 27

6.9 툴 개발 28

IV 아키텍처 **29**

V 데이터베이스 **31**

VI 부하 테스트 **33**

VII 게임 데이터 **35**

VIII 게임 시스템 **37**

Part I

글쓰기

Chapter 1

어떤 글을 써야 하나?

1.1 쉬운 글

글은 쉬어야 한다. 쉬운 글은 누군가와 말하듯이 쓰는 글이 좋다. 말하듯 편하게 얘기하면서 그 뜻이 분명하다면 기본은 된 글이다.

1.1.1 한자어를 피한다

애매하다와 같은 단어도 한자어이다. 우리는 한자 문화를 오래 유지했기에 피할 수는 없다. 적절하게 잘 선택하여 90대 할머니도 알만한 단어를 써야 한다. 또는 아홉살 조카에게 존대말로 말하듯이 쉽고 편안하게 써야 한다.

1.1.2 짧게 쓴다

한 문장이 길어지면 쓰는 사람도 한 문장을 쓰는 동안 처음부터 끝까지 잘 말하기가 어렵다. 또 길면 문법적으로도 오류가 생기기 쉽다. 그래서 글을 짧게 하는 것이 대체로 좋다.

1.1.3 핵심을 쓴다

개념의 이해나 사용법 등 전달하고자 하는 대상을 정하고 필요한 결과를 얻는 가장 효율적이고 필수적인 것들을 쓴다.

그렇게 하려면 내가 먼저 잘 이해해야 한다. 이해의 폭을 넓히기위해 자료를 수집하고 정리하고 연습하여 훨씬 더 넓고 큰 이해를 갖고 있어야 오류 없이 쉽게 전달할 수 있다.

1.2 명확한 글

기술과 기술을 익히는 것에 관한 책들을 쓸 예정이므로 내용이 명확해야 한다. 명확하다는 뜻은 글을 읽었을 때 그 의미하는 바가 하나로 일관된다는 뜻이다. 문장의 의미가 하나로 둘이 아니다.

또 여러 문장으로 풀어서 설명할 때 그 흐름이 일정하여 목표하는 전달에 집중할 수 있고 그 결과로 만들어질 의미들이 일관되어야 한다.

1.2.1 코딩하듯이

가장 명확한 글쓰기는 코딩과 수학의 증명이다. 코딩은 특정 기능을 오류 없이 서술하는 과정이고 수학은 특정 사실을 오류 없이 다른 사실들에서 이끌어 내는 과정이다.

단언(assert)을 코딩에서 증명할 때 사용하듯이 글쓰기도 단언들이 붙어서 스스로 증명되듯이 명확하게 만드는 방법을 찾고 적용하도록 해보자.

1.2.2 대상을 명확하게 안다

대상을 먼저 잘 이해하고 간결하게 잘 설명할 수 있어야 한다. 이해가 우선이다. 개발자로서 경험이 많지만 정리 안 된 영역이나 세부 사항이 많으므로 그런 부분들을 공부해서 정리한다.

이것이 아마도 가장 큰 경쟁력 중 하나가 되지 않을까?

1.3 재미있어야 한다

재미있는 기술서적은 쓰기가 어렵다. 감성적인 대상이 아닌 논리적인 대상에 대한 이해와 활용을 목표로 하는 모든 글들이 재미있는 경우는 많지 않다.

프로그래밍 책이 재미있다는 뜻은 "아하" 하는 순간이 많다는 뜻이다. 수학 문제를 풀면서 "아하"하고 기쁨을 느꼈던 순간들, 큰 기능을 설계하고 구현해서 통합하고 잘 동작할 때 "와"하고 기쁨을 느꼈던 순간들을 제공할 수 있어야 한다.

그렇게 하려면 동기부여가 필요하다. 어디에 쓰이는지 어떤 일화가 있었는지 적고 작은 목표들을 달성하는 길을 안내한다.

1.4 반복하여 개선한다

한번에 좋은 결과가 있기를 바라면 안 된다. 인공지능도 자연지능도 모두 끊없는 반복을 통해 만들어지고 그 규모가 클수록 더 좋은 결과를 만든다.

보고 또 보고. 연습하고 또 하고.

1.5 함께 한다

이 과정을 다른 사람들과 함께 해야 한다. 누구일지는 모르겠으나 함께 한다. 적절한 시기에 같이 하면서 피드백을 받아야 한다.

Chapter 2

예쁜 문서 만들기

2.1 라텍 잘 다루기

LaTeX으로 문서를 작성한다. 따라서, 잘 다룰 수 있어야 한다. Overleaf와 같은 미리 잘 준비된 환경이 아닌 직접 환경을 구성해서 작업한다. 잘 다룰 수 있도록 하려고 그렇게 한다.

LaTeX은 아주 많은 패키지들이 있다. 인연이 닿는 좋은 패키지들을 우선 쓴다. 그리고 점점 더 나은 방법들을 찾는다.

2.2 폰트

예쁜 문서의 핵심은 배치와 폰트이다. 폰트가 더 큰 영향을 준다. 목적에 맞는 폰트를 기능단위로 몇 개 나누어 고르고 일관되게 사용한다.

2.3 배치

배치(레이아웃)도 문서를 예쁘게 하는데 큰 기여를 한다. 제목만 잘 구성해도 멋진 문서처럼 보인다. 줄간격이나 페이지에 담기는 내용의 조밀도도 중요하다.

코드가 많으므로 코드 내용을 깔끔하게 잘 보여줄 수 있어야 한다.

우리나라 교과서들이 한 페이지에 여러 정보를 구조화하여 보여주는데 탁월하다. 이 부분도 참고한다.

Chapter 3

LaTeX

3.1 빈 페이지

왜 챕터 내용이 안 보이는 줄 알았는데 빈 페이지들이 하나씩 있어서 찾지를 못 했다.

3.2 한글 폰트 지정

fc-list 명령으로 폰트 설치 폴더들을 볼 수 있다. 그 중에 truetype 폴더 아래에 ttf 파일을 복사하면 쓸 수 있다.

setmainhangulfont 명령으로 지정하면 된다. 서로 다른 폰트로 조판하면 더 좋은데 나중에 살핀다.

3.3 프로젝트 구조

내용이 많을 경우 장 단위로 파일을 구성해도 관리가 어려울 수 있다. 따라서, 내용이 많은 책을 쓸 경우는 section 단위까지 나눌 필요가 있다.

프로젝트마다 각 장의 크기가 다르므로 잘 선택해서 진행한다. 코드가 많은 프로그래밍 책 특성 상 절 단위로 파일을 나눠야할 것 같다.

Part II

내용

Chapter 4

무엇을 쓸 것인가?

4.1 안내를 한다

게임 서버 프로그래밍에 필요한 내용들을 찾아서 이해하고 사용할 수 있는 연습을 할 수 있도록 안내를 한다.

연습은 반복을 한다는 뜻을 함께 갖는다. 이해도 반복해서 하면 더 깊어지고, 사용도 반복해서 하면 더 능숙하게 된다.

4.2 정리를 한다

안내를 위해서는 잘 알아야 하므로 찾고 관계를 이해하고 내용을 정리해서 안내를 준비한다.

두 과정이 잘 맞물리면 점점 좋은 내용으로 발전할 수 있다.

4.3 실제 내용들

게임 개발을 위해 알아야 하는 내용들은 많다. 따라서, 모든 내용을 정리하고 연습으로 만들기는 어렵다. 그러므로 스스로 찾아 갈 수 있는 훈련이 되도록 하고 단단한 기초가

되는 부분들은 확실하게 정리하면 좋다.

실제 개발을 하면서 필요로 했던 내용들과 필요하지만 아직 명확하게 정리되지 않은 것들을 중심으로 정리하고 쓴다.

- 라이브러리
- 통신
- 데이터베이스
- 분산 처리
- 게임 데이터
- 게임 시스템

Part III

라이브러리

Chapter 5

연습 준비

연습을 위한 준비와 연습을 위한 방법을 정리한다.

- ide 준비하기
- vcpkg
- doctest 사용법
- 디버깅
- 코드 읽기

ide는 visual studo와 visual c++로 한다. 일반적으로 쉽게 접근 가능하므로 이를 기준으로 하고 vscode + cmake + clang은 가볍게 언급한다. 이 쪽으로 진행하는 사람도 있을 수 있다.

doctest는 단위 테스트로 기능을 꼼꼼하게 살펴서 연습하는 방법으로 설명한다. 단위 테스트와 assert에 해당하는 CHECK로 연습하게 한다. TDD를 가볍게 언급하고 현실적으로 적용 가능한 수준에 대해 정리한다.

Chapter 6

라이브러리 목록

게임 서버 개발에 필수적인 라이브러리들로 이미 사용했거나 사용할 것들을 정리하고 연습으로 만든다.

6.1 표준 라이브러리

- container
- string
- memory
- file
- thread
- lock
- atomic
- chrono
- regex

6.2 boost와 신규 라이브러리

std에 없는 것들을 언급할 필요가 있다. 찾아서 연습할 수 있도록 한다.

- boost multi_index
- redis skiplist

6.3 tbb

이외에 tbb에 있는 컨테이너들도 중요하고 유용하다.

- concurrent_queue, concurrent_map

6.4 메모리 관리

메모리 관리 라이브러리들도 중요하다.

- tcmalloc
- mimalloc

6.5 통신

통신과 관련한 기본 라이브러리들이다.

- asio
- cpr
- sodium
- crypto++
- flatbuferrs

6.6 데이터 관리

데이터 관리를 위한 라이브러리이다.

- RapidCsv
- nlohmann json
- nanodbc
- redis

6.7 공간과 물리 처리

공간과 물리 처리를 위한 라이브러리이다.

- recastdetour
- bulletphysics
- box2d

6.8 메트릭

메트릭 라이브러리들이다.

- prometheus
- grafana
- exporter 들

6.9 툴 개발

툴 개발을 위해 필요한 라이브러리이다.

- imgui

Part IV

아키텍처

Part V

데이터베이스

Part VI

부하 테스트

Part VII

게임 데이터

Part VIII

게임 시스템