1. 대략적으로 1달 정도의 시간이 흘렀습니다.

1달의 시간 동안 느낀점과 간략한 소감을 작성해주세요.

부가적으로 앞으로의 포부 또한 같이 작성해주시면 감사하겠습니다.

-한달동안 새로운 것들을 배우면서 머리는 아프지만 새로운 것들을 알게 되는게 재미있었습니다.

팀과제도 학교에서 하던 팀플보다 보람찼습니다.

이대로 열심히 해서 과정을 다 마무리하면 어디든 들어갈 수 있다고 느껴져서 열심히 잘 해보겠습니다.

​

2. 언어를 가리지 않고 Domain Driven Design을 수행하여 얻을 수 있는 이점들에 대해 기술하세요.

-모든 코드들이 도메인별로 잘 나누어져있어서 새로운 것을 추가하고 코드를 이해하고 에러를 고치기가 쉽다.

​개발할 때 복잡성을 제어하기 좋다.

3. C/C++ 에서 Test 목적으로 GoogleTest를 사용하였습니다.

TDD(Test Driven Development)를 통해 얻을 수 있는 이점들에 대해 기술하세요.

​

-테스트를 미리 함으로써 문제를 먼저 발견해서 해결할 수 있다.

필요하지 않은 것들을 먼저 쳐낼 수 있다.

4. DDD(Domain Driven Design)과 TDD(Test Driven Development)의 관계에 대해 기술하세요.

​

-DDD를 하면서 생기는 구조들을 Test 해봄으로써 그 구조가 제대로 만들어지는지 알아볼 수 있다.

5. 팀원들과 협업을 진행 할 때 중요하다 생각하는 요소들을 작성하고

각 요소들에 대해 조금 더 상세하게 기술하세요.

​

-협업을 할 때 역할을 나누고 자신이 해야하는 부분이 무엇인지 확실하게 아는것이 중요하다고 생각한다.

해야하는 것을 확실하게 알아야 자신에게 팀에게 필요한 것이 무엇인지 알 수 있고 협업이 진행될 수 있다.

6. 함수 포인터 테이블을 사용하는 이유에 대해 IoC 관점에서 기술하세요.

​

-함수 포인터 테이블로 기능을 나누어서 고치기 쉽고, 이름으로 호출할 수 있게 만들어서

코드를 알아보기 쉽다.

7. virtual method를 사용하는 이유에 대해 기술하시오.

​

8. 협력형 과제를 진행하는 상황 혹은 회사에서 업무를 진행하는 상황에서

갑자기 대응하기 어려운 문제를 마주하게 되었습니다.

이런 경우 여러분들은 어떤 형식으로 문제에 대응 할 것인가요 ?

어떻게 대응 할 것인지 상세하게 기술하세요.

-협력하는 팀원들과 문제에 대해서 회의를 하고, 해결방안을 찾아 볼 것입니다.

​

9. Backlog를 작성함으로서 얻을 수 있는 이점들에 대해 기술하세요.

-Backlog를 작성함으로서 어느부분이 어떤 의도로 작성되었는지 쉽게 알 수 있다.

작업이 얼만큼 진행됐는지 알아보기 쉽다.

​

10. 여러분들은 이미 Board (게시판) 을 Console UI와 함께 구현해봤습니다.

이 때 여러분들이 각자 분업하여 만들었던 작업들이 존재합니다.

여러분들이 각자 분업하여 만든 내용물들이

다른 사람들의 결과물과 원활하게 결합하지 못하여 폐기처분하는 경우도 있었습니다.

혹은 끔찍한 혼종이 탄생하는 경우도 있었습니다.

이런 현상이 나타나게 된 근본적인 이유에 대해 기술해봅시다.

​

-분업해서 만든 결과물을 결합하는데에 중요한 요소들을 충분한 회의 없이 만들면

다른사람의 결과물과 결합하지 못하고 폐기하게 될 것 입니다.

11. 기능 단위로 Backlog 를 작성하는 경우 어떤 문제들이 발생 할 수 있는지 기술해봅시다.

-Backlog 만으로는 무엇을 하려고 작성된 코드인지 알아보기 어려워집니다.

코드해석에 시간이 사용됩니다.

​

[ 복합 문제 12 ~ 20 ]

12. 속도 Entity를 작성합니다. (복잡도를 낮추기 위해 2D 로 제한)

​https://github.com/youngchan4/SDC-Comprehensive-Evaluation/tree/main/1%EA%B8%B0/first/youngchanhwang/comp12

13. DB에 속도 정보와 등록 시간을 저장 할 수 있는 테이블을 작성합니다.

​CREATE TABLE speedDB (

speed DECIMAL(10,3) NOT NULL,

reg\_date datetime(6) DEFAULT CURRENT\_TIMESTAMP(6)

);

14. DB에 접속 할 수 있는 싱글톤을 생성합니다.

​

15. 소수점 형태의 랜덤 sleep 기능을 구현합니다.

​

16. 10번 정도 루프를 돌면서 0.3 ~ 1.0초 사이의 시간 이후 랜덤 속도 정보를 DB에 저장합니다.

​

17. 가속도(Acceleration) Entity를 작성합니다 (마찬가지로 2D)

​

18. 속도 정보를 토대로 가속도 정보를 산출합니다.

​

19. 속도 정보를 토대로 전체 이동 거리를 계산합니다.

​

20. 속도 정보를 토대로 현재의 변위 벡터를 계산합니다 (초기 위치 [0, 0] 으로 가정)

​

21. 현재 실시간으로 분석되는 영상 시스템을 사용하고 있습니다.

이 영상 시스템은 촬영하는 구간에 사람이 몇 명 있는지 확인 할 수 있습니다.

이러한 시스템을 특정 시간에 사람들이 붐비는 구간에서 사용하고자 합니다.

영상 시스템의 목적은 버스 정류장에 사람이 몇 명 있는지 판정하려고 합니다.

어떻게 DB에 데이터를 저장해야 스토리지를 효율적으로 사용 할 수 있을까요 ?

​

-정해놓은 수 이상의 인원수 변동이 있을때 DB에 데이터를 저장한다.

22. 포인터가 필요한 이유가 무엇인지 기술하세요.

-포인터를 사용하면 포인터에 저장된 주소로 바로 접근해서 사용할 수 있기 때문에

함수나 변수에 쉽게 접근할 수 있게된다.

​

23. 모든 함수들이 자신만의 개별적 공간인 Stack을 사용합니다.

함수들끼리 서로 Stack 을 공유하지 않는 이유에 대해 기술하세요.

​

[ 복합 문제 24 ~ 32 ]

24. C언어로 Product (고유번호, 가격, 상품명, 상품 상세 설명) Entity를 작성합니다.

​https://github.com/youngchan4/SDC-Comprehensive-Evaluation/tree/main/1%EA%B8%B0/first/youngchanhwang/comp24

25. 해당 Entity와 관련한 사항들이 잘 동작하는지 확인하기 위해

GoogleTest를 통해 테스트를 진행합니다.

​

[ 복합형 DDD 문제 26 ~ 30 ]

S(변위) = V\_0(초기 속도) \* t(시간) + 0.5 \* a(가속도) \* t^2(시간 제곱)

V(속도) = V\_0(초기 속도) + a(가속도) \* t(시간)

​

위 공식은 고등학교 물리학 시간에 배우는 등가속도 법칙입니다.

시간에 따라 값들이 변하는 시간 함수 형태라면 위의 식들에 미적분이 필요합니다.

(일단 우리는 미적분을 적용하는 문제를 풀지는 않을 것입니다)

​

우주에서 레일건을 쏴서 적군의 전함을 공격하는 시나리오를 가정하겠습니다.

아주 이상적인 환경이라 가정하고 계산을 진행합니다.

​

전자기력을 이용하여 물체를 가속시키기 때문에 우주에서도 사격이 가능합니다.

레일건 함포의 길이는 110m 입니다.

탄환은 10m 구간에서 가속하여 발사됩니다.

고도의 과학 기술로 인해 전자기력에 의한 가속도는 500만 m/s^2 로 일정하다 가정합니다.

​

26. 문제에서 거론된 함포가 쏘는 탄환의 종단 속도를 계산 할 수 있도록 Backlog를 작성합니다.

​https://www.notion.so/eddi-robot-academy/EPIC-2-0dd919fe74254670a8c1c0426eee781d

27. 이에 대한 테스트 코드를 작성합니다.

​https://github.com/youngchan4/SDC-Comprehensive-Evaluation/tree/main/1%EA%B8%B0/first/youngchanhwang/comp26

28. 작성한 Backlog를 토대로 탄환의 종단 속도를 계산합니다.

​

29. 탄환이 포신의 종단에 도달하기까지 걸린 시간을 계산 할 수 있도록 Backlog를 작성합니다.

​

30. 이에 대한 테스트 코드를 작성합니.

​

31. 작성한 Backlog를 토대로 탄환이 포신의 종단까지 걸린 시간 값을 계산합니다.

​

32. 적군의 전함까지 떨어져 있는 거리는 대략 10000 km 입니다.

함포를 발사했을 때 적군의 함정에 착탄되기까지 걸린 시간을 계산해 봅시다.

​

33. 처음에는 Domain 분리가 필요 없다 생각하였던

특정 Entity 내부의 어떤 멤버 변수 (필드) 가 있습니다.

시간이 지남에 따라 시스템이 커졌고 위 내부 변수가 수행하는 작업의 복잡도가 높아지고 있습니다.

여러분들이라면 이 녀석을 어떻게 처리하겠습니까 ?

​

[ 복합 문제 34 ~ 35 ]

이 문제를 통해서 C언어로 Request, Response 객체의 free()를 크게 신경쓰지 않아도 됩니다.

​

34. C언어로 Request 객체를 전송하고 이 객체를 받아서 변환 이후

알아서 자동으로 객체가 free 되도록 코드를 작성해봅시다.

​

35. C언어로 응답 객체를 만들고 이를 요청했던 부분에 전달합니다.

Response 객체를 사용하면 자동으로 객체가 free 되도록 코드를 작성합시다.

​

36. Valgrind를 구동하여 Memory Leak이 발생하지 않는지 최종적으로 체크합니다.