

1. 대략적으로 1달 정도의 시간이 흘렀습니다. 1달의 시간 동안 느낀점과 간략한 소감을 작성해주세요. 부가적으로 앞으로의 포부 또한 같이 작성해주시면 감사하겠습니다.

-학교에서 배우는 정도의 난이도라 생각하고 왔지만 DDD방식에 대해 듣고 어셈블리어에 대해 들으면서 내가 배운 것은 쉬운것이라는 것을 느꼈으며 또한 새로운 것을 배우는 것에 대한 재미 또한 있었습니다. 앞으로 여러 언어와 협력하는 방법을 배우면서 함께 프로젝트를 완성해봤으면 좋겠습니다

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

-객체 지향에 기반한 재사용성, 확장성 그리고 유지 보수의 편리하다

-메서드 이름만으로 무엇을 하는지 알 수 있을정도로 이해하기 쉽다.

-커스텀 가능성이 높고, 다양한 통합 및 확장기능을 제공합니다.

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

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

-코드를 테스트하고 오류 및 버그를 확인하기 때문에 유지 관리와 확장이 용이한 코드를 생산할 수 있게 된다.

-테스트를 실행하며 오류를 초기에 잡아낼 수 있기 때문에 장기적으로 보면 개발 속도를 높일 수 있다.

-다른 사람이 gtest에 테스트한 코드를 보고 코드에 대해 더 잘 이해할 수 있다.

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

-TDD를 DDD를 토대로 나온 모델과 연결고리들이 적합한지 빠르게 테스트 해볼 수 있으며 DDD를 확장하거나 결합하기 쉬워진다.

5. 팀원들과 협업을 진행 할 때 중요하다 생각하는 요소들을 작성하고

각 요소들에 대해 조금 더 상세하게 기술하세요.

-에자일보드를 작성해야한다 이유로는 해당 팀원이 무슨 작업을 하는지 잘 알 수 없고 나중에 중복된 결과가 나올 수 있기 때문이다.

-서로가 알 수 있는 단어를 선정하여 코드를 작성해야 한다 예를 들어 대충 변수를 **a**, **b**식으로 코드를 만들었을 경우 다른 팀원은 이 코드가 무엇을 하는 것인지 잘 모르며 작성자 또한 나중에 기억이 나지 않을 경우 해당 코드를 이해하는데 많은 시간을 써야되기 때문이다.

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

-함수 포인터 테이블을 사용하면 해당 함수를 다른 곳에서도 활용할 수 있으며 객체 간의 결합도를 낮추고, 코드의 재사용성과 유지보수성을 향상시킬 수 있기 때문이다.

7. **virtual method**를 사용하는 이유에 대해 기술하시오.

-부모 클래스에서 만든 함수를 자식 클래스에서 해당 함수를 재정의 할 수 있도록 허용하기 위해서입니다.

-**virtual**없이 함수를 만들었을 경우 자식 클래스에서 해당 함수를 이용하더라도 부모 클래스에서 만든 결과대로만 출력하게된다.

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.

-우선 팀원한테 해당 문제에 대한 해결법으로 아는 방법이 있나 물어볼 거 같습니다.

그러나 이 방법으로도 해결하지 못했을 경우에는 해당 문제를 인터넷 검색이나 해당 책을 찾아 볼 거 같습니다.

9. **Backlog**를 작성함으로서 얻을 수 있는 이점들에 대해 기술하세요.

-팀원들이 무엇을 하는지 쉽게 파악이 가능하며 작업의 현재 상태 또한 파악이 가능하다

-**backlog**는 작업의 우선순위, 상태, 소요 시간 등과 함께 관리되며, 프로젝트를 효율적으로 추적하고 계획하는데 도움을 줍니다.

10. 여러분들은 이미 **Board** (게시판) 을 **Console UI**와 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.

-에자일보드에 **backlog**를 구성하기 않고 그냥 각자 만들었고 구체적으로 소통 또한 하지 않았기에 때문이다.

-**gtest**를 사용하지 않았기에 코드를 테스트해보기가 힘든 문제가 발생하였고 이로 인해 나중에 결합할 때 에러가 발생할 경우 많은 시간을 소요하게됩니다.

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

- 시간이 지날수록 여러 복잡한 작업이 진행됨에 따라 해당 **backlog**의 목적이 불분명해지게 됩니다 그러므로 해당 **backlog**를 보더라도 무엇을 위해 만들어졌는지 이해하기 힘들어지고 다른 팀원이 봐도 만든 목적을 잘 모르게 되기 때문에 **backlog**에 대한 소통 또한 힘들어지게 됩니다.

12~20.

<https://github.com/Amicitia00/SDC-Comprehensive-Evaluation/commit/43f6c39f63f91884b5c7e08804aafe966e19ffd0>

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 **DB**에 데이터를 저장해야 스토리지를 효율적으로 사용 할 수 있을까요 ?

-

22. 포인터가 필요한 이유가 무엇인지 기술하세요.

-간결하고 효율적인 표현과 처리가 가능하고 더 빠른 기계어 코드를 생성할 수 있으며,
복잡한 자료 구조와 함수의 쉬운 접근이 가능하다.

23. 모든 함수들이 자신만의 개별적 공간인 **Stack**을 사용합니다.

함수들끼리 서로 **Stack** 을 공유하지 않는 이유에 대해 기술하세요.

-**stack**에는 함수 호출 시 전달되는 인자와 되돌아갈 주소값도 쌓인다

-이를 **stack memory**라 하며 쓰레드마다 할당된다.

-쓰레드가 생성되고 시작될 때마다 고유한 스택 메모리를 가지기에 다른 쓰레드 스택에 접근이 불가능하다

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 ?