슬더스가 되고싶었어

이번 개인 프로젝트는 콘솔환경에서 개인이 만들고 싶은 게임을 만드는 프로젝트였습니다.

저는 제가 구현이 가능하다고 생각이들은 범위 안에서 최대한 모든 기능을 구현하려고 했습니다.

해당 게임장르의 게임을 주제로 만든건 정말 심플하고 간단한 이유였습니다.

가장 최근에 들은 수업이 LinkedList를 통한 객체관리와 데이터취급

그리고 Dictionary를 통한 데이터 전달 이였습니다.

그날 강의를 듣고 든 생각은 아. 이거 카드게임에서 덱이나 인벤토리 혹은 아이템 파운더 등으로

사용하면 대박이겠다 하고 뇌리를 스쳤습니다. 그래서 평소 TCG 장르를 좋아하던 저는 나중에

slay the spire 일명 ‘슬더스’ 같은 장르의 게임을 만들고 싶다고 다짐을 했었지요.

그게 생각보다 조금 이르게, 가능한 형식 선 안에서 구현해본게 이번에 만들어본

STS가 되고 싶었어 입니다.

만드는 중 막히고 고생한 부분은 정말 많았지만, 단순히 강의를 수강하면서 타자 연습을 하듯 받아 적은 문장들과 제가 만들고 싶은 게임을, 구현 하고싶은 기능을 만드는 것은

나 자신에 대한 원동력이 되어주었기에 일주일 이라는 시간이 쏜 살 같이 흐르듯

몰두 할 수 있었습니다,

서론이 길었지만 지금부터 STS가 되고 싶었어 의 이야기를 시작하겠습니다.

처음의 구상은 역시 막 배운 LinkedList의 활용 이였습니다.

Player용 LinkedList를 하나 구성한뒤 Card라는 클레스로부터 값을 받아와

‘덱’을 구성했습니다. 하지만 덱 이란 가져오고, 소모하며, 순환하는 요소가 전부 들어가야했습니다.

해당 방법을 어떤 식으로 구현 할 지 정말 많은 고민을 하였는데

플레이어가 카드를 얻거나, 제거하는 행위를 통해서 덱을 수정하는건 기본이 되는 DeckList에 할당을 해준뒤

게임에 돌입하여서 쓰는 덱은 DeckList로부터 받아오는 TempDeck으로 구성하여서 카드를 받아왔습니다.

매턴 TempDeck으로부터 최상위 4장을 받아오며 더 이상 뽑을 카드가 존재 하지 않을 시

사용한 카드를 다시 셔플해 덱에 넣는다. 굉장히 심플한 구성이였습니다.

TempDeck의 최상위 4장을 다시 HandList로 할당하였고

카드의 발동 조건, 카드의 타입, 카드의 밸런스 등을 구성 하였고 기본틀을 잡아갔습니다.

카드 게임에 가장 기초가 되는 덱을 구성한 뒤 에는

UNIT이라는 하나의 부모 클래스로부터 정보를 받아오는

Player와 Monster이라는 두개의 클래스를 구성했습니다.

하지만 두개의 클래스에 차이점을 두었는데 그건 바로

Monster을 사용해 생성한몹들에게는 디버프를 받았을 때 N턴뒤에 끝난다 등의 요소를 넣기위해서

ATK부분을 get set 하는게 아닌

get으로만 받아오는 OrignalATK라는 변수를 따로 구성후에 카운터가 0이 되면 OrignalATK 로 돌아가는 방식으로 디버프를 구현하였으며

Player의 부분에서는 플레이 하는 유저가 과도한 스톨링(적을 죽이지 않고 자원이나 체력등을 체워가는 행위)를 하여 게임의 재미가 떨어질까 우려하여

player쪽에도 OrignalHp로 get만 받아오고 OrignalHp의 값을 Hp가 받은뒤에

하나의 층계가 종료 됄 때 Hp가 OriganslHp 보다 높을시에 OrignalHP(HP의최댓값)으로 바뀌도록 구성하였으며 특정 조건에따라(몬스터처치, 휴식) 등으로 OrignalHp를 늘리는 요소도 구현했습니다.

몬스터와의 전투는 하나의 메서드로 압축한뒤에 대상이 됄 몬스터를 할당하는 식으로 진행하였으며 몬스터의 카드 디자인이 어려워 이부분을 유저가 해당층계에서 가질 평균적인 카드의공격값과 유사하게 몬스터의 공격력을 설정한뒤 변수값을 할당해서 매턴 유저와 합을 진행후에

이기는 쪽이 행동을 하는 식 으로 구성 했습니다.

다음에는 몬스터의 몬스터의 카드를 할당, 몬스터의 체력,코스트,공겨력 등에따른 행동방식을 구현 하고싶습니다.

또한 애를 먹은 부분이 화면이 전환됄 때 마다

어차피 새로 출력 하는데 그냥 clear치고서 작성작성 만들 때 편해서 좋겠네 “ 라는 오만한 생각을 했었는데 이런 커다란 사고를 친거를 나중에 가서 정말 후회 했습니다.

콘솔 클리어를 난발한 제 게임은 패에 있는 카드를 선택할때마다 화면이 점멸하였고

이 부분은 가시성에서 정말 큰 마이너스로 돌아왔습니다.

해당부분을 고치기위해 매서드를 전부 조사하면 Console.Clear가 들어가는 부분을 다 삭제한후

다음 행동이 들어갔을 때 컴퓨터가 그릴 부분을 덧그리고 필요 없는 부분을 공백으로 다시그려 지우는 식으로 다시 구성 했습니다. 모든 매서드를, 함수를 다 조사한 덕 분에 그나마 점멸하는 현상을 최소화 해서 가시성을 늘렸습니다. 오늘의 교훈을 뼈저리게 기억하며 앞으로 구성할 모든 코드에 이유없는 Consol.Clear을 난발하지 않을 좋은 경험을 했습니다.

작업을 하면서 오늘 작업할 내용을 사전에 작성후 코드를 구성하면서

문제가 생기는 부분을 그때 그때 기록하면서 추후에 볼 수 있도록 지우지 않고 일자별로 분류를 해놨는데 이런 습관이 오늘날에 회고록을 작성하는데 정말 많은 도움을 줬습니다.

기술적인 부분에서는 미흡한 작품 일 지라도 오늘 저는 좋은 습관을 들이는 방법과

잘못된 습관을 고칠 수 있던 정말 중요하고 재밌는 경험이였습니다.