**구현과정 관련 설명**

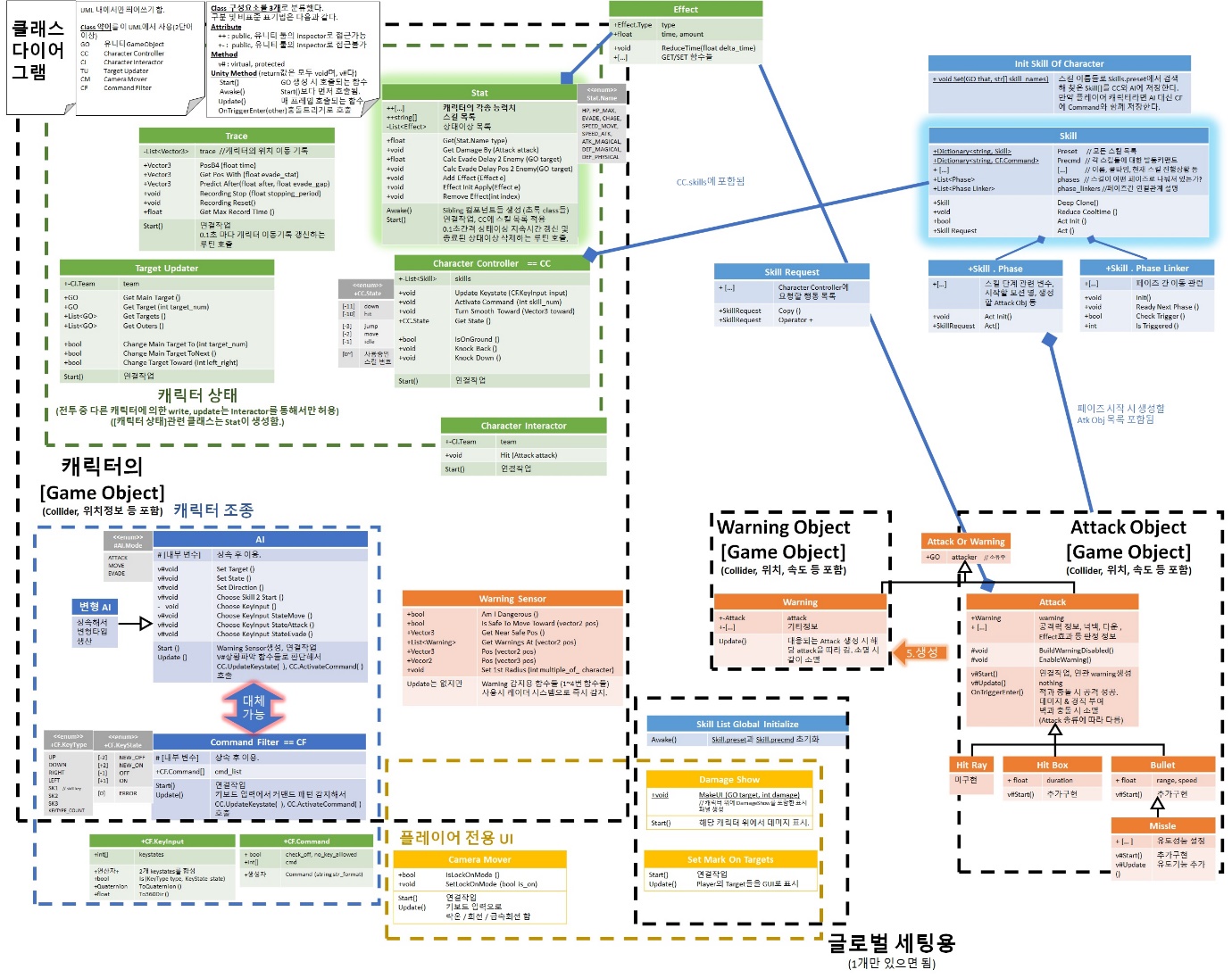
본 개인프로젝트는 C#, 유니티를 전혀 모르는 상태에서 배우면서 진행한 프로토타입입니다. 고등학교 시절 기획이 현실성 있는지 테스트하기 위한 목적으로 만들어졌으며, 약 20명의 인원들을 대상으로 30분가량의 유저테스트를 2회 실시했습니다.

완전히 완성하지는 않았지만, **기대한 결과는 다 이뤘기에 미완성인 채 종료**한 프로젝트입니다.

**현실적으로 구현 가능한지 조사**하고,

**유저가 재미있게 즐길 수 있는지 조사**하고,

**실제 게임을 완성하지 않더라도 대규모 게임을 위한 기반을 다져본다**는 목표는 충분히 이뤘습니다. 이 프로젝트를 종료한 뒤 현재는 다른 기획의 프로토타입을 제작중입니다.



**프로그래머로서 어려웠던 점**은 **NPC AI의 회피기동 구현 작업**입니다.

저는 AI가 **상대방의 준비동작을 보고 회피, 날아오는 공격을 보고 이동경로를 정해 회피**하기를 원했습니다. 문제는 크게 2가지입니다. “**다가오는 위협을 어떻게 감지할 것인가? 회피경로를 어떻게 잡을 것인가?**“ 입니다.

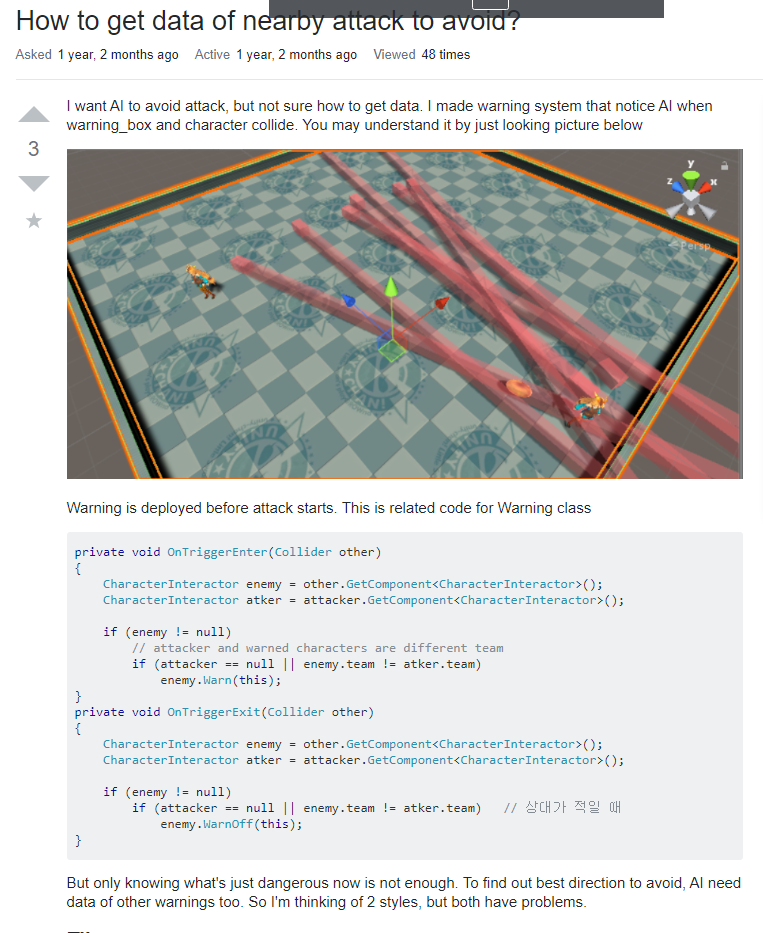
주변의 위협적인 공격을 감지하는 것은 어렵지 않습니다. 일정 거리 이내에 공격이 들어오는지 감지하면 끝입니다. **문제는 탄환을 감지하더라도, 그 탄환이 나를 향해 날아오는지**, 아니면 전혀 다른 엉뚱한 곳으로 이동하는지 확인하기가 어렵습니다. 더군다나 상대방의 선딜레이 동작만 보고 판단하는 것은 더욱 어려운 일입니다. **상대가 사용하는 스킬이 어떤 스킬인지 하나하나 판단하는 것은 무식하고, 정확히 나를 조준하고 있는지**, 사거리가 닿는지를 판단하는 것도 효율적인 작업은 아닙니다.

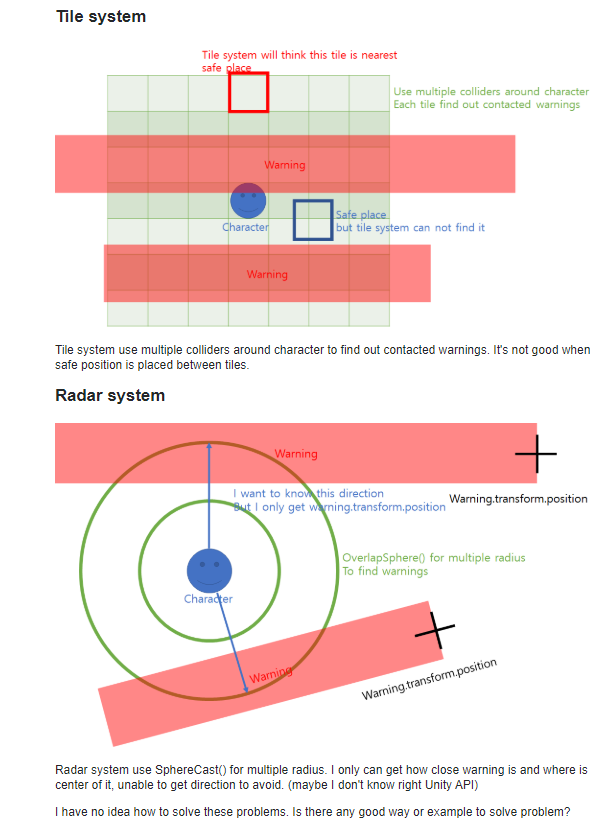
더 큰 문제는 어느 방향으로 피할 지 결정하는 것입니다. 해당 공격판정의 형태와 이동속도, 지속시간을 바탕으로 위험구역을 만드는 것은 가능하지만 굉장히 어렵고 힘들고 무거운 작업입니다. 실시간으로 여러명의 AI가 행동하기엔 끔찍한 일입니다.

저는 이를 Warning Box를 구현해 해결했습니다. 모든 캐릭터는 공격 스킬 사용 즉시 공격예정인 장소에 warning box를 생성합니다. 이렇게 할 경우 **선딜레이 감지와 탄환감지를 동일한 프로세스**로 처리할 수 있습니다.

**회피경로를 구하는 것도 쉽습니다**. 자기 주변을 n\*n개의 타일로 구분한 뒤 각 타일별로 warning이 있는지 판단해 안전한 타일로 도망가면 됩니다. 평상시엔 타일로 쪼개서 모든 상황을 감지할 필요 없고, 자기 주변 2m 내에 warning이 주어지는지 1번만 체크하면 연산 부하도 줄일 수 있습니다.

아래는 개발당시 며칠 고민하다가 stack overflow에 질문했던 글의 스크린샷입니다.





|  |  |
| --- | --- |
| **기능** | **참고할 파일** |
| 클래스 다이어그램 등의 설계도 | [다이어그램]폴더 |
| 프로토타입 시연 영상 | Watchme |
| 프로토타입 작동 | play\LucidDream.exe(low 퀄리티 구동 추천) |
| 조작법 | Play\readme조작법 |
| 소스코드 및 유니티 프로젝트 | [LucidDream]폴더 |
| 유저테스트 결과 | [유저테스트]폴더 |