

# 기술과 도구 개발에 관해

---

2018년 5월 17일에 어느 스튜디오에서 발생한 일들이다.

- 빌드가 안 됨
- 스킴 관련 판정 버그
- 아이템 관련 버그
- 일본어 한자가 있으면 로그가 남지 않음

## 빌드 이슈

---

빌드 이슈는 데일리 빌드를 전체 프로젝트에 대해 실행하고 있으면 발생하지 않을 수도 있는데 관리할 브랜치가 많아지면 어려운 측면이 있다. 퍼포스를 사용하고 github의 브랜치 / 머지 흐름을 사용하고 있는데 git 처럼 분산 형상 관리 도구가 아닌 경우 잘 맞지 않는 면이 있다. 책임의 주체가 없고 비 프로그래머인 아티스트, 기획자들이 함께 형상의 변경에 들어오기 때문이다.

git은 크고 많은 바이너리 데이터 처리에 적합하지 않다고 알려져 있다. 측정을 통해 확인한 바는 아니지만 소스 관리에 대부분 최적화된 구조를 갖고 있고 git 도 그런 역사를 갖고 있기 때문에 그렇게 추정한다.

만약 git이 아트 애셋을 포함해서 처리할 수 있다고 가정한다면 위와 같은 이슈가 발생했을까? 빌드 이슈는 코드 이슈인데 왜 아트 애셋 때문인가 생각해 볼 수 있다. 아트애셋을 포함하여 기능 단위까지 브랜치가 있다면 완결되기 전에 병합될 일이 없기 때문이다. 기능 단위로 브랜치를 하려면 아주 많은 양의 바이너리 데이터의 브랜치와 병합이 매우 빨라야 한다. 이 기능이 아직 git에서 지원이 원활하지 않은 부분이다.

## 스킴 관련 판정 버그

---

서버에서 판정 단위를 정확하게 하기 위한 수정이 있었다. 만약 이 기능이 원래 일반적이면서 매우 빠른 오래 연구되고 다양한 환경의 테스트가 끝난 기능이었다면 고생할 필요가 없었다.

서버의 공간 처리를 충돌, 판정, 물리, 길찾기를 포함하여 지원하는 라이브러리가 있다면 발생하지 않았을 문제이다.

## 일본어 한자 로그

---

윈도우 서버에서 다국어 지원은 UTF16LE로 한다. 로그 파일의 인코딩에 따라 변환이 이루어져야 한다. 여기서도 기반에 대한 준비가 없으면 고생을 하게 된다는 점을 여실히 보여준다.

C++ 기반의 서버에서 UTF8 기반으로 내부 문자열 처리, 로그, DB, 통신이 모두 처리되고 테스트 된 라이브러리를 확보해 두었다면 문제가 없었을 것이다.

## 왜 못 했는가?

---

개별 프로젝트 진행은 가장 빠른 해결 방법을 택하게 된다. 왜냐하면 인력 투입이 증가하는 상태에서 기술 개발에 투여되는 시간은 비용 증가를 배수로 늘리기 때문이다.

프로젝트 경험이 전혀 없다면 요구 사항을 몰라 별도의 기술과 도구 개발이 가능하지 않다. 하지만, 어느 정도 경험이 쌓이면 이를 찾을 수 있다. 개별 프로젝트를 진행하고 나면 개선해야 하거나 새로운 아이디어가 많이 생긴다.

게임 개발과 기술 개발이 순환되도록 해야 한다.

