

“ 웰컴 미스터 맥도날드 ”
Welcome Mr. McDonald

(주)로커스홀딩스 손노리 게임사업본부

NEW팀 프로듀서 서관희

2001-11-03

소개말

- 발표자 Profile

- 1993~2000년까지 프로그래머, 디렉터, 팀장을 개념 없이 중복 작업하며 PC,아케이드,콘솔에서 게임을 출시 함.
- 2000년 중반부터 프로그램 팀장에서 프로듀서 직책으로 전환 화이트데이 팀을 운영.

소개말 (계속)

- 강연 내용 Overview

- 화이트데이 개발에서 겪은 여러 가지 문제점에 대한 정리와 개선을 위한 꿈틀거림을 소개.
- 게임 개발에서 출시까지 몇 가지 사항들(인력, 리소스, 일정, 개발 외적인 요소)에 대한 개인적인 의견을 피력함.
- 강의 자료만 읽어 볼 때도 이해 될 수 있도록, 가능한 서술형으로 내용을 정리했음.

Why? 웰컴 미스터 맥도날드

- 원제는 “라디오의 시간”이라는 일본 영화.
- 우여곡절이 많았으나 끝내 완성을 이루는 모습이 우리의 경우와 너무나 비슷하여 다른 의미로 감명 깊게 봄. (게임 개발자라면 꼭 한번 보시라!)



[게임 개발 현장에서
흔히 겪는 장면]

희망찬 도전

- 팀으로 활동하던 손노리가 1998년 주식회사로 발전하면서 의욕 넘치는 개발 의지를 가지고 3가지 타이틀(강철제국, 악튜러스, 화이트데이) 동시에 추진.
- 그러나, 당시 개발 인원은 13명.
- 2D, Half 3D, Full 3D 게임이며 장르도 모두 다른 게임을 개발하면서 당시 매니저급 개발자는 각자의 담당 업무를 병행 해야만 했다.
- 실로 엄청난 도전...

화이트데이 개발 목표

- 자체 3D 엔진 개발과 3D게임 개발 노하우 획득
- 네트워크 게임 개발 기술 확보
- 버그로 인한 오명에서 벗어나고 진부한 이미지 개선

(반성 : 개발자 마인드의 개발 목표만 있었고 사업적인 목표는 막연했다.)

화이트데이 개발 목표 (계속)

- 자체 3D 엔진 개발과 3D게임 개발 노하우 획득
 - 바닥부터 시작한 개발 과정 속에서 엔진 구입에 대한 필요성과 장단을 절실히 느꼈음. (이에 대해서는 추후 논의)
 - 2D→3D로 전환되면서 개발이 3배나 어렵다는 것을 알게 되었음. (특히 1인칭 게임의 경우)
 - 기획, 그래픽, 프로그래밍 등등 전 파트에 걸쳐 어떠한 작업에 대한 프로세스가 2D에 비해 많아져 개발의 탄력이 쉽게 붙지 않음을 알게 되었음.
 - 자작 툴(In house tools)의 개발 시에는 스케줄에 많은 신경을 써야 하며 툴 전담 개발자의 필요성을 절실히 느끼게 되었음.

화이트데이 개발 목표 (계속)

- 네트워크 게임 개발 기술 확보

- “멀티 플레이 게임 개발” 결정은 화이트데이 팀의 부담을 2배로 안겨주는 결과였다는 것을 뒤늦게 알게 됨.
(반성 : 결국 동시에 2개의 다른 게임을 만들어야 했음.)
- 온라인 게임과 다르게 멀티 플레이 게임(특히 FPS 게임)들이 현재의 모습을 가질 수 밖에 없는 것을 이해하게 됨.
- 현재 알파 테스트 준비 중이고 화이트데이와 동시에 출시 하지 못한 것을 매우 안타깝게 생각하고 있지만 2001년 중반부터 본격적으로 시작한 것에 비해 대단히 빠른 개발 속도라 생각 됨.
- “존 카맥”님을 존경하게 됨.

개발 초기의 치명적 실수

- 낙관적인 생각만으로 개발에 들어 갔으며 전혀 체계적이지 않은 초기 기획 작업으로 아무런 디자인 문서도 없이 1년간 허송세월을 보냈음.
- 개발 초기 프로듀서가 3번 바뀌는 등 일련의 상황들은 책임 소재가 불명확했던 것에서 시작했으며 개발 초기에 책임과 권한을 명백히 하지 못한 탓.
- 이는 창립 초기 경영진(본인 포함)의 비합리적인 결정 방식으로 인한 것이 었음을 깨닫고 이를 개선 하고자 부단히 노력하고 있는 중.

개발 단계 (1년차 : 1998. 10 - 1999. 6)

- 초기의 인력은 6명 (기획2, 프로그램2→3, 그래픽2).
- 아이디어 수준의 기획 단계.
- 프로토타입 수준의 엔진 개발.

x: -97 y: -1 z: 67
Contents: -1
FPS: 29 LC: 0 TC: 0 DP: 355



3월 13일반 학교에서 벌어지는 피의 축제: 화이트데이
Original Copyright by Sonnori Co., Ltd. 1999. WhiteDay and WangReal Engine are trademark of Sonnori Co., Ltd.



응.. 근처에서 작은 여자에 못봤니?

개발 단계 (2년차 : 2000. 1 - 2000. 10)

- 개발 인력이 20여 명으로 증가.
- 게임 기획이 새롭게 시작됨.
- 엔진의 스펙도 대부분 확정되었음.
- 멀티 플레이 개발을 아직 시작하지 못함.



개발 단계 (3년차 : 2001. 1 - 2001. 9)

- 대화 이벤트 대폭 수정. 그래픽 보강.
- 사운드 전담 인원 총원.
- 개발자들의 스트레스가 가장 심했던 때. (봄)
- 점차 피드백이 빨라지면서 개발에 박차를 가함.
- 완성도 문제로 여름 시즌을 보내고 9월말 발매.
- 멀티MOD는 현재 개발 중.

고무줄 데드라인

- 초기에 우리의 능력을 과신한 개발 기간 설정.
- 장르 특성상 여름에 출시해야 한다는 부담감에 무리한 스케줄 단축과 이를 반복하는 일이 연속됨.
- 해본 적이 없는 기술적 항목들로 인해 스케줄 추정 불가능한 경우가 많이 있었음.
- 보통 예상한 기간의 2배를 잡아 보기도 했지만 때때로 3배까지 걸리는 경우가 많이 있었다.
- 데드라인에 대한 스트레스는 증가하고 잦은 발매 연기로 개발팀의 사기가 극도로 저하 되었다.

외부 자원과 기술

- 엔진에서 개발 맡기에 사운드 라이브러리인 Miles Sound System을 사용하기로 결정.
- One Site로 어느 정도의 비용이 들었으나 사운드 전담 프로그래머가 없는 상황에서 안정성 확보를 위해 선택.
 - 주의 : Miles를 알맞게 요리하는 것도 약간 어려울 수 있음.
- 홍보용 일러스트의 경우 외주로 진행.
 - 일정을 조절하기 쉬웠으나 계약된 그림 외의 추가 일러스트가 많이 부족하게 됨.
- 음악의 경우 오리지널 곡이 아닌 기존의 국내 음악들 중에서 게임에 가장 적합하다고 생각한 것을 라이선싱하여 사용. (이에 대해 추후 논의)

음악 라이선싱

- 음악을 사용하기 위한 라이선싱의 과정이 매우 번거롭고 불안정함. (국내의 저작권 개념이 매우 애매함.)
- 한국 음악 저작권 협회에 지급 하는 “게임 음악 사용료”는 정액이지만 저작권자(작곡가)와 저작인접권을 가진 음반사(또는 발행사)와 개별 연락해서 추가 계약을 해야만 했다. (예상치 못한 비용 초과가 많다.)
- 게임의 성격에 맞는 음악을 미리 선곡할 수 있어서 좋았고 반응도 좋았기 때문에 다행이었지만 예산 책정이 어렵다.

외부 업체와의 관계

- 인텔 코리아와 긴밀한 관계를 유지.
 - 개발 장비 지원으로 펜티엄4 프로세스에 옴티마이징된 엔진을 개발할 수 있었다.
- 퍼블리셔인 “위자드 소프트”가 일정에 대해 이해를 해주고 믿어 주었기 때문에 많은 도움이 되었다.
 - 하지만 QA나 마케팅에서 앞서 이끌어 주는 모습이 없어 아쉬웠음.

프로토 타입과 마일스톤

- 개발 초기부터 1~3달 간격으로 내부 시연회를 자주 열어 사내 다른 팀과 경영진에게 소개 했고 개발의 Milestone (이정표)으로 삼았다.
- 엔진과 툴의 초기 버전이 나왔을 때 프로토 타입 제작에 들어가 게임의 도입부를 구현.
- 임시 코드로 범벅된 것이 였지만 개발팀과 외부인에게 게임의 대한 이해를 높이는데 큰 역할을 했음.
- 횡수를 거듭 할수록 되던 기능이 안 되거나 게임 진행 가능한 범위가 오히려 축소 되는 경우가 있어 엔진 자체 개발의 어려움을 통감 했다.

프로토 타입의 중요성 (계속)

- 그 뒤 1년간 엔진이 계속 업데이트 되었지만 스토리상 구현된 내용은 동일했다. (2000년 ECTS와 2001년 가을, 정식 데모버전이 거의 비슷한 분량까지만 진행 가능했음.)
- 하지만 속의 내용은 상당 부분 개선 되어가고 있었다.
- 프로토 타입 제작으로 게임이 출시되기 1년 전부터 동일한 인터페이스를 유지하게 되어 이에 대한 다양한 개선을 시도해 볼 수 있었다.

품질 관리 (Quality Assurance)

- 애초에 QA 담당 정직원 또는 외부 업체에 의뢰할 계획이 었지만 국내에 QA에 대한 개념이 아직 미흡하여 원하는 사람/업체를 찾지 못하였다.
- 앞으로 QA 분야에 대한 체계적인 연구가 선구자들에 의 해 이루어지기를 희망하며 국내 게임의 전반적 수준 향상 이 요구될수록 QA에 대한 중요도가 높아지리라 예상한 다.
- 우리는 결국 아르바이트 베타테스터를 상주시켜 테스트를 했으며 “박스 라이브”라는 Bug Tracking System을 이 용해 문제를 DB화시키려 노력했다.

박스 라이브 – Bug Tracking System

- 몇 개의 게임 출시 때마다 반복되는 버그 테스트에 대한 효율적 방법 모색의 결과.
- Bugzilla(GDMag Korea 11월 호에도 나옴)를 모델로 우리 목적에 적합하게 Php+MySQL로 개발.
- 출시 전까지 1,100여 개의 버그가 리포팅 됨.
(첨부 그림 참고 바람)
- 추후 타 개발사를 위해 시스템 공개를 계획 중. (<http://developer.sonnori.com>)



팀 내/외 커뮤니케이션

- 내부 게시판.

- 프로젝트 초기부터 Staff 서버를 두고 내부 팀별 게시판을 운영. 또한, 최신 기획 문서에 누구나 접근하기 쉽게 만들기 위해 노력함.
(공개된 JS 게시판 개조해 사용.)
- 강제로 모든 직원의 IE 첫 페이지를 Staff 게시판으로 지정.

- 공유 캘린더.

- 전체 인원이 공유하는 웹상의 캘린더. 회사 내 공유 이벤트 (외부 업체 내방, 생일, 휴가, 세미나 일정) 를 쉽게 알 수 있도록 제작.

- Popcorn - 사내 채팅 프로그램.

- Light하면서 유용한 채팅 프로그램. Broad Casting 기능으로 실시간 전체 공지에 유용함.

- 워크 페이지 - 사내 업무 지시 및 주간 업무 일지 기능.

워크 페이지

- 첨부된 그림을 참고 바람.
- 바쁜 업무 중인 사람을 방해하지 않고 사소한 일을 부탁하려는 의도에서 개발 시작.
(Post-It을 모니터에 붙이는 것에서 착안)
- 점차 주요 업무 배분, 주간 업무 일지 기능, 같은 파트의 다른 사람들 일정 확인 등의 용도로 활용.
▶ 팀 내 커뮤니케이션에 크게 일조.
- 역시 MySQL과 Php로 자체 개발 & 개선.
▶ 프로듀서, 프로그래머, 기획자에게 Php를 배우게 하면 여러모로 쓸모 있고 재미 있는 것들을 해 볼 수 있으니 권장 함.

적절한 회의 방법 모색

- 적절한 회의 방법을 찾기 위해 다양한 시도를 함.
- 팀원이 20명을 넘어서면 전체 회의 의미가 사라짐.
 - 의견 제시 없는 분위기가 되거나 난상 토론이 되기 쉬움.
- Chief 회의를 주초에 열어 핵심 사항을 논의 후 전체 회의는 공지 형식으로 함.
- 매 회의 시간은 1시간을 넘지 않도록 노력.
- 개발이 멀티와 시나리오로 나뉘면서 회의를 별도로 진행 함.

지친 개발자 달래 주기

- 팀원들의 퍼포먼스가 떨어지는 4가지 경우
 1. 여자친구가 없어 외롭다.
 2. 여자친구가 생겼다.
 3. 여자친구와 싸웠다.
 4. 여자친구와 헤어졌다.

☪ 참고로 화이트데이 팀원은 100% 남자 T_T

게임 개발의 주요 포인트는 사람이기 때문에
개발자를 위한 정책을 펴기 위해 노력.

개발자들이 바라는 환경

- 의욕 있고 능력 있는 동료와 훌륭한 팀워크.
- 프로젝트에 대한 확고한 비전과 이를 이끌어 갈 수 있는 리더.
- 좋은 개발 시스템과 주변 환경.
- 자신의 능력에 맞는 업무와 이에 대한 적절한 권한과 대우를 받는 것.
- 일에 자신을 희생하면서 무리 하는 게 아닌 자기 발전을 위한 기회가 되길 바란다.

→ 물론, 우리도 아직 많이 부족하다. 안심 하시길...

잠재적 위험 요소인 개발자

- 기초가 부족하고 노력하지 않는 개발자.
- 시킨 업무만 하는 개발자.
- 업무에 집중하지 못하는 개발자. (메신저 & 웹서핑)
 - ➔ 언뜻 보기에 문제가 없어 보일 수도 있기 때문에 관리의 사각지대에 있기 쉬움.
 - ➔ 주변 개발자의 의욕을 저하 시키므로 심각한 문제.
 - ➔ 집중 관리와 단속 또는 결단을 내야 한다.

엔진 자체 개발에 대한 개인적인 의견

- 엔진을 자체 개발 하느냐 구입 하느냐는 게임의 성격(장르, 플레이 방법, 플랫폼)과 스펙(기술적 요구 사항, 프로젝트/팀 규모)에 달렸다고 생각.
- 자체 개발하면서 엔진 구입에 대해 뼈저리게 느꼈지만 자체 개발 경험 없이 엔진 구입을 한다면 동급의 문제들에 직면하게 될 것임.
- 엔진 구입이 개발자를 줄이는데 도움이 된다고 생각하면 오산.
- 엔진 구입은 핵심 개발자가 엔진에만 매달리는 문제를 해결 할 수 있고 게임에 집중 할 수 있는 여지를 만들어 준다. (단, 소화해 낼 수 있는 개발자를 갖추었다는 가정하에서.)

카피 프로텍트

- SafeDisc2를 적용.
- 출시 전 우려했던 대로 발매 하루 만에 복사 퍼짐.
- CD-Key를 적용 했으나 네트워크 게임이 아니기 때문에 공유가 심해짐.
- 업데이트 제작 시 마다 프로텍트 적용해야 하기 때문에 신속 대응이 어렵고 패치 용량 증가의 원인이 됨.
- 프로텍트로 인해 실행되지 않는 것을 버그로 비난.
- 개발사의 피해에 대한 적절한 보상이 없음.

고유 번호 인증제 도입

- CD-Key 공유와 부적절한 복제를 막기 위해 도입.
- WindowsXP에서 사용 한다는 것과는 무관하며 개인 사생활 침해의 목적으로 사용되지 않음.
- 네트워크를 통한 온라인 인증은 인터넷에 항상 연결되지 않은 유저는 플레이 불가능하기 때문에 시스템 환경을 Key로 하는 인증 방식 도입.



고유 번호 인증제 도입 (계속)

- 인터넷이 되지 않으면 여전히 불편 하지만 패치 후에만 적용되기 때문에 큰 문제는 없었고 효과가 있다고 판단 됨.
- 대략 구입자의 30% 정도가 패치와 인증을 받음.
- 추후, 자동 응답 시스템(ARS)를 이용한 인증 개발 등으로 계속 발전 시켜 보고 싶음.

PC패키지의 살길 (개인적 관점에서)

- 개인적으로 싱글 플레이/패키지 게임 개발을 좋아 하지만 시장 상황의 개선 여지가 불투명한 상황에서 개발 의욕이 상실 됨.
- 네트워크 인증 기반의 멀티 플레이와 플레이어의 기록이 보존되는 시스템을 갖추지 않고서는 패키지로서의 상품성이 인정 받지 못하는 현실이 안타까움.
- 콘솔(비디오 게임기)처럼 비교적(?) 복사가 없는 플랫폼으로 전환 또는 남다른 온라인 개발사로 거듭나는 모습이 필요 할 듯.
- 하지만, 감동적인 스토리를 지닌 게임을 언젠가 꼭 만들어 보고 싶기에 그때까지 개발력 향상을 위해 노력 하겠음.

Thanks

감사합니다.

끝.