



(주)네오플

# 新 야 구

이 상 록  
(주)네오플 개발팀장

GAME REPORT



신 야구 개발 리포트에서는 기획과 그래픽 그리고 개발과 관련 엔진의 3가지 테마를 중심으로 글을 전개할까 한다.

## 신 야구 개발 배경

맨 처음 신 야구라는 온라인 야구 게임을 생각 했을 당시 수많은 야구 게임 중에서 신 야구의 포지션과 야구라는 스포츠를 온라인 게임으로 어떻게 재구성할까가 기획 단계에서 고민하게 된 첫 번째 과제였다.

기획 초기만 해도 스포츠게임의 온라인화는 보기 드문 상태였으며 스포츠 게임은 인기를 얻기 힘들다는 편견이 있었다.

그러한 편견과 환경 속에서도 야구 본래의 게임성과 국내 프로야구를 통해 어느 정도는 대중화된 스포츠라는 매력에 제한된 유저 층이지만 온라인 야구 게임으로서 충분히 야구 게임의 재미를 줄 수 있고 흥행도 할 수 있다고 판단하게 되었다.

결국 이러한 이유와 더불어 온라인에서 서비스되고 있는 야구 게임이 전무한 상태에서 회사내의 야구를 좋아하는 개발자 중심으로 온라인 야구게임에

대한 갈망과 함께 개발자들이 만들어 즐기자는 기분으로 신 야구 개발이 시작 되었다.

결과론이지만 신 야구 오픈 시점은 캐주얼 온라인 게임의 성장과 더불어 다양한 장르에 대한 유저들의 요구와 맞물려 온라인 스포츠 게임의 관심이 신 야구와 함께 높아졌다.

## 신 야구 게임 컨셉

앞 장에서도 잠시 언급 했지만 온라인 야구 게임은 아니지만 수많은 야구 게임이 나와 있는 상태이며 거의 모든 게임이 리얼 야구를 표방하고 있으며 장기간의 개발 노하우와 함께 상당한 완성도를 갖추고 있는 게임들이 많았다.

야구라는 게임을 처음 만드는 입장에서는 기존의 많은 리얼 야구 게임과 똑같은 컨셉으로 가게 된다면 승부가 힘들다고 판단했으며 신 야구의 특징인 온라인 야구 게임의 특징을 최대한 부각시키며 캐주얼 온라인 게임답게 누구나 쉽게 접근해서 야구라는 스포츠에 대한 접근성을 최대한 이끌어낼 수 있는 컨셉을 생각하게 되었다. 최종적으로는 만화같은 친근한 아케이드성이 강한 캐주얼 야구 게임을 컨셉으로 내세우게 되었다.

만화 같다는 건 현실적인 야구 구현의 부담을 줄일 수 있었으며 온라인 게임에 어울리는 캐주얼 게임 분위기와 야구에 대한 접근성을 끌어내는 효과가 있다고 판단하였다.

그렇다고 비현실적인 야구 룰에 야구와 동떨어진 게임을 만들수는 없는 것으로 기본적인 야구 룰은 그대로 유지하여 야구의 재미를 떨어뜨리지 않고 간편하게 야구를 즐길 수 있도록 심플한 게임 진행과 조작이 가능하도록 기획되었다.

## 기획적 특징

신 야구의 전반적인 기획적 특징은 기존의 야구 게임이 가지고 있지 못한 부분의 구현에 있다. 일반적인 리얼 야구의 실제 선수를 내세워 정해진 능력치로 게임을 진행하는 야구 게임과 달리 선수 육성이라는 육성 개념이 있기 때문에 다양한 형태의 선수 육성에서 나오는 팀 특성을 가진 팀들과의 온라인 대전과 육성한 선수를 수 많은 유저들 대상으로 트레이드하는 등 야구 게임 이외의 온라인 게임이기 때문에 가능한 야구 외적인 재미 요소를 강조하고 있다.

이러한 요소들은 실제 야구세계에서도 일어나고 있는 일들이며 마치 자신이 구단주가 되어서 경기를 하고 선수를 육성하고 트레이드하는 야구로 즐길 수 있는 모든 재미적 요소를 담고 있어 궁극적인 게임의 목표인 최고의 구단주가 되는 게임이 되기도 한다.

그 외 게임 플레이를 들여다 보면 야구 게임의 핵심인 투타의 맛을 살리는데 최대한 중점을 두고 개발

하였다.

타격감과 투구감을 살리기 위해 동기화를 맞추는 부분이나 타격 타이밍, 물리 적용 등의 고민을 통해서 야구의 기본적인 치고 던지는 맛을 살리려고 하였다.

그리고 야구 게임의 복잡한 컨트롤과 물을 최대한 쉽고 심플하게 풀려고 컨트롤 키를 방향키와 3개의 선택키로 최소화하였으며 주루 플레이, 타격 등 최대한 쉬운 조작을 표방하였다.



## 신야구 3D

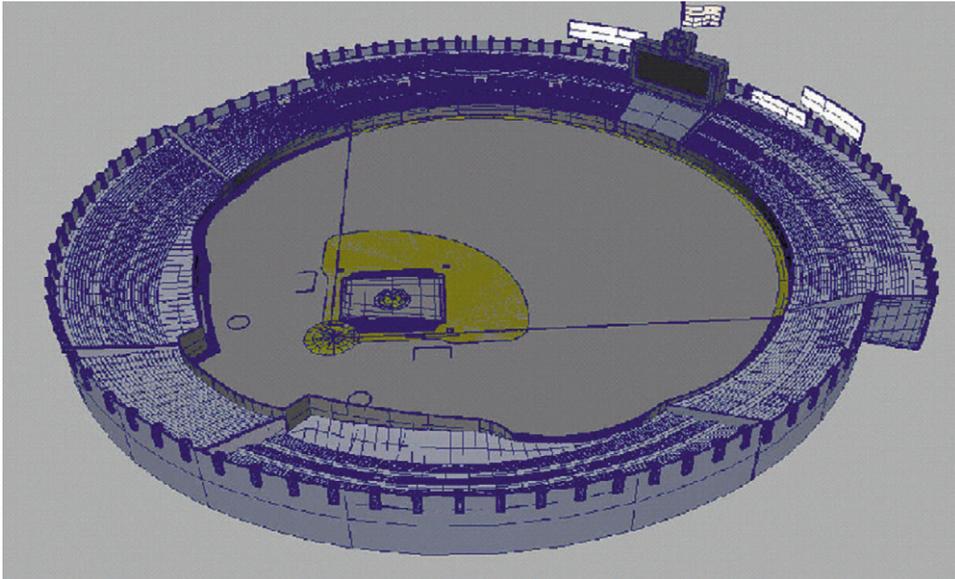
신 야구는 만화같은 야구게임을 표방한 캐주얼 게임이다. 선수캐릭터와 구장 및 기타 요소들이 3D로 만들어져 있으나 게임상에서는 2D 기반으로 적용되므로 다른 3D게임처럼 폴리곤 수나 맵 소스의 크기에 연할 필요는 없었다. 그러나 2D게임 특성상 콘텐츠의 양이 많아질수록 이미지의 양이 기하급수적으로 늘어나기 때문에 워크 플로우의 효율성과 데이터 관리 능력이 요구되었다. 3D오브젝트들은 모두 마야를 사용해 만들었다.

## 구장

구장은 실제 야구장에 근거하여 작업하였다.

구장내 구조물들은 모두 3D로 작업한 뒤 투수부, 타자부, 구장부를 렌더링 하였고, 관중은 차후에 2D로 합성하였다.

항상 보여야 하는 부분이라 면을 아끼지 않고 사용했으며, 튼적인 느낌을 살리기 위해 램프셰이더 (Ramp Shader)를 중심으로 여러 셰이더들을 사용하였다.



## 일반선수

### 모델링 및 텍스처링

렌더된 이미지를 쓰는 것이므로 폴리곤 보다는 보다 적은 용량으로 부드러움을 표현할 수 있는 넵스가 선호되었다. 물론 오브젝트의 특성에 따라서 넵스, 폴리곤, 섭디비전을 고루 활용하였다.

셰이딩은 구장과 마찬가지로 램프 셰이더를 중점적으로 사용하였다. 또한 마야에 내장된 벡터 렌더 (Vector Render)로 외곽선을 만들어 합성하였다.

초기 버전의 신 야구는 홈팀, 어웨이팀 두팀으로만 이루어져서 홈팀은 붉은색과 흰색 조합, 어웨이는 푸른색과 검정색 조합의 유니폼을 착용하고 있는 형태였다.

그러나 KBO 라이선스를 획득하고 아바타 시스템을 적용하기로 하게 되면서 기존의 플랫폼이 바뀔수 밖에 없게 되었다

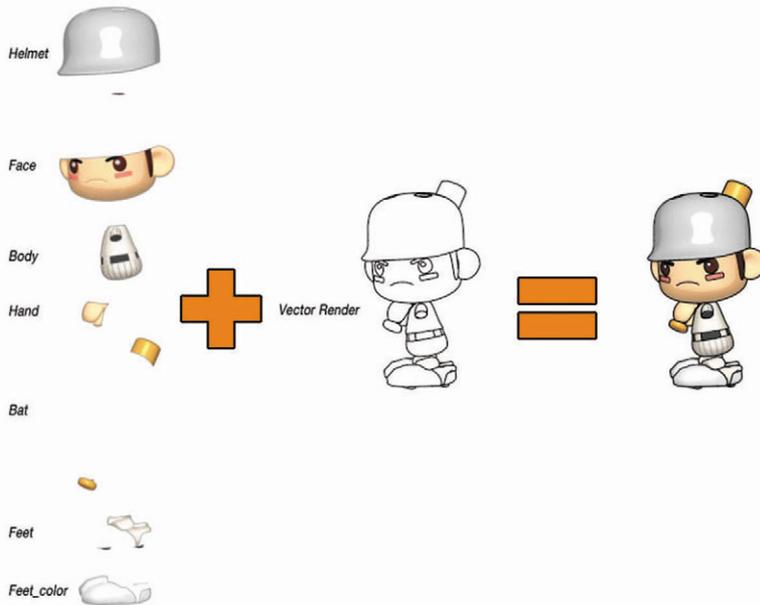
현존하는 8개구단과 자유팀, 네오플팀을 합쳐 총 10개의 기본 유니폼이 필요했는데, 한 팀씩 렌더링할 경우 그 양이 너무 많아지기 때문에 헬멧과 신발 부분에 흰색 셰이더를 입혀서 렌더링을 걸고, 구단별로 그부분에 색을 지정해서 프로그램상에서 입혀주는 방식을 썼다.

물론 하나의 캐릭터를 통째로 렌더를 거는 방식도 파트별로 다 나누어 거는 것으로 바뀌게 되었다.

렌더 횟수는 많아졌으나 적은 용량으로 10개 구단을 해결할 수 있고, 또 이렇게 나누어 거는 것이 아바타 시스템의 기초가 되는 것이었다. 원하는 부분만을 렌더링 하기 위해서 해당부분 이외에는 유즈백 그라운드(use background) 셰이더를 사용하였다.

벡터 렌더는 컬러이미지 위에 멀티플라이(multiply)로 덧입히는 방식이었으므로 캐릭터 전체를 한번에 렌더링하고 그 이미지를 모든 파트에 다 적용시켰다.

컬러 이미지와 벡터 이미지를 합치는 톨은 자체적으로 제작해서 사용하였다.



아바타 시스템에 들어갈 유니폼이나 글러브, 배트 등의 장비는 미리 준비해 놓은 씬에서 셰이더만 바꾸어 다시 렌더를 거는 형식으로 진행되었다.

SD이기 때문에 왜곡되는 부분도 있겠지만 기존 구단들의 유니폼도 최대한 비슷해 보이도록 맵을 만들고, 자유 팀과 네오플 팀도 유니폼을 따로 디자인해서 만들어주었다.

파트별로 떨어져 있어서 여러 가지 조합으로 다양한 캐릭터를 만들 수 있다는 장점을 살리는 데는 성공했으나, 렌더 시간과 후 작업에 많은 시간이 소요되고, 데이터 양이 많아진다는 이유로 다양한 체형의 캐

릭터를 선보이자는 안은 실현될 수가 없었다.

## 애니메이션

초기에는 실제 사람처럼 본 세팅을 하고 키 애니메이션을 했으나, 아바타 시스템으로 전환될 때 캐릭터 셋업도 변화가 있었다. 팔다리가 없는 캐릭터에 굳이 본 세팅을 할 필요가 없다고 판단하여 그룹을 만들고 해당그룹의 어트리뷰트에 키를 주는 방법으로 전환하였다. 발 부분에만 이례적으로 조인트를 바인딩 했는데, 이는 발이 꺾이는 부분을 표현하기엔 위의 방법이 부적합 했기 때문이다.

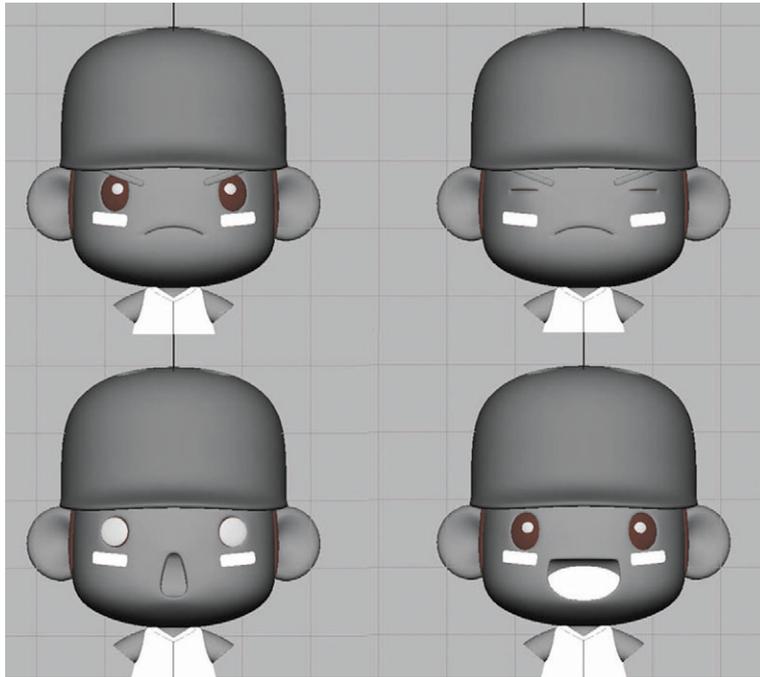
바뀐 캐릭터 셋업은 예상대로 더 좋은 효율을 가져왔다. 애니메이션을 잡는 시간도 줄고, 키 값을 클립으로 익스포트하여 다른 캐릭터에 적용하는 것도 훨씬 간단한 셋업으로 가능해졌다.



키를 잡는 과정에서 약간의 애로사항이 있었는데, 신 야구의 캐릭터는 팔다리가 없고, 허리와 골반이 움직이는 것도 아니어서 동작이 상당히 제약되어 있다는 것이었다.

스포츠 게임의 다이내믹 함을 표현하기에 좀 힘들었고, 캐주얼 게임이기에 귀여운 요소도 갖추어야 했는데 다이내믹함과 귀여움 사이의 정도를 맞추는 것도 쉽지 않은 작업이었다. 캐릭터의 표정은 블렌드 셰입(Blend Shape)으로 타겟을 지정하여 만들었다.

기본적으로 세 가지의 표정(눈 깜빡임, 놀람, 기쁨)을 갖게 하였다.



## 게임 적용

렌더링한 컬러 이미지와 벡터 이미지를 합치고 그 파일을 묶어서 각각 하나의 img 파일로 만들고, 동작 별로 ani파일을 만들어 게임에 적용시켰다. 이 부분은 모두 자체 개발 툴로서 쉽게 해결할 수 있었다.

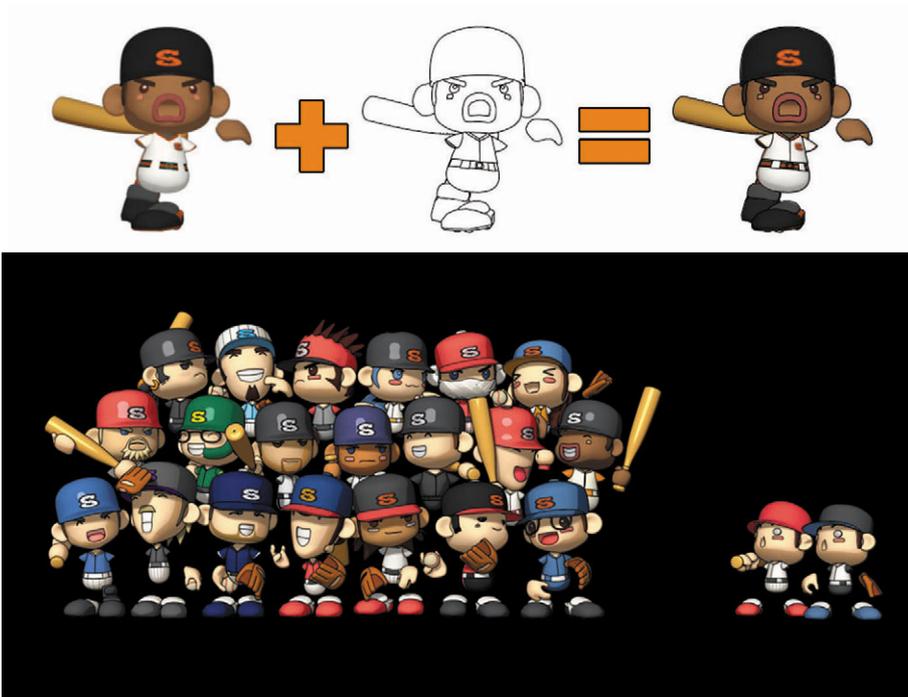
그러나 이미지 데이터가 점점 방대해지면서 용량을 줄일 솔루션이 필요했는데, 결국 32비트 이미지를 nmg라는 16비트 이미지로 압축하게 되었다. 물론 용량에서의 이득은 얻을 수 있었으나 이미지의 질이 떨어지는 것은 감수해야 했다.

## 영웅캐릭터

기획 단계에서 총 20명의 용병 캐릭터가 제시되었는데 7명은 가상의 캐릭터이고 나머지 13명은 기존 선수를 기반으로 만든 캐릭터였다.

용병은 각자 고유한 동작을 갖는데, 그것을 처음부터 일일이 다시 키를 잡지 않고 일반선수와 셋업을 동일하게 맞추고 일반선수의 키 값을 임포트해서 편집하는 형식으로 진행하였다. 이런 방식의 워크플로우는 작업 간소화는 물론 나중에 ani파일을 만드는 시간도 줄일 수 있게 되는 등 매우 효율적이었다.

용병에 관한 아쉬움이 있는데, 용병이 다른 유니폼이나 장비를 착용하는 것은 불가하다는 것이었다. 역사나 시간과 데이터양의 문제였는데, 용병 캐릭터가 일반 선수와 다른 크기와 다른 셰입을 갖기 때문에 현실적으로도 불가능한 일이었다.



## 신 야구 엔진의 특징

신 야구 엔진의 제작은 2.5D의 특징을 최대한 살리고 LOW TEEN들도 어려운 RENDERING 설정 등이 없이 쉬운 게임 접근을 용이하도록 한다. 또한, 웹을 통한 빠른 설치를 위한 리소스 통합 관리 등을 기본 전제로 하여 제작을 하게 되었다.

### Make It Wonders! Make It Fast!!

#### [그래픽 엔진 NEngine 의 특징]

1. Hardware - accelerated , scenegraph - driven frame work  
고정 함수 파이프 라인을 통한 하드웨어 가속을 지원.

구장, 투수, 타자의 각각 다른 씬들의 노드들을 관리하고 제어하기 위한 Scenegraph를 기반으로 한다.

## 2. Full Support for sprites – translate, animate, flip and rotate and zoom

제작 기간의 최소화를 위한 DIRECTX 의 DXSPRITE를 기반으로 하여 제작하였다.

## 3. Ultimate number of fully dynamic rendering Layers

여러 계층의 다양한 렌더링 레이어를 지원한다.

## 4. Camera rotation, zoom

2.5D 이미지의 왜곡 등이 없는 특성에 맞는 카메라의 회전, 줌을 지원하여 다양한 화면 구성을 지원하도록 한다.

## 5. Rendering filtering

NEAREST, LINEAR등의 기본 샘플링과 필터링을 지원하며, 텍스처 블렌딩을 지원한다.

## 6. DDS 파일 포맷 지원

DirectX에서 쓰이는 텍스처와 같은 포맷을 사용함으로써 텍스처 생성 후 파일에서 로딩한 데이터를 텍스처의 서피스로 복사해 넣을 때 DDS 파일 포맷을 지원한다.

일반 포맷에 비해 상대적으로 비디오 메모리를 아낄 수 있고, 많은 개수의 텍스처, 큰 크기의 텍스처를 낮은 코스트로 사용할 수 있을 것 같았으나, 리소스가 32비트이고, graduation이 많으며 또한, 기타 이유로 DXTC FORMAT을 위하여서는 너무 많은 낭비를 하게 되었다.

텍스처 크기가 257×258 이미지의 경우 512×512 로 사이즈가 늘어나게 되므로, 이를 막기 위한 방법이 필요하였다. 이 경우는 256×256, 1×256, 256×2, 1×2로 텍스처 4개를 생성한 후에 화면에 출력할 때 붙인 조각을 잘라서 붙이는 방법으로 개선하였다. DDS 파일 포맷의 특성상 주위 픽셀에 따라 색이 많이 왜곡되고 흐려지는 현상이 존재하고 있어, 지금도 대안을 준비 중이다.

## [물리 엔진]

야구의 특성에 맞추어서 투수가 던지는 공의 다양한 구질 처리와 다양한 타구 처리를 위한 발사체 등의 기본적인 물리엔진을 목표로 제작을 하게 되었다.

Fully Rigid – Body Dynamics simulation

Physics – based object mounts

Kinetics 지원

다양한 구질 처리를 위한 동역학 기반의 메소드를 제공한다.

ProjectTile 지원

다양한 타구 처리를 위한 발사체 기반의 메소드를 제공한다.

### [사운드 엔진]

기본적으로 신야구는 스포츠 게임이기 때문에 조합형 해설을 지원하여야 하기 때문에, 상당한 리소스가 소요될 것으로 예상되었다. OGG 지원 – 리소스 절약 및 높은 음질 지원을 위한 OGG 파일을 기반 제작하였으며, 각각의 음원을 조합하여 부드러운 해설처럼 보이기 위한 버퍼 캐쉬 관리 및 자원 관리를 담당하는 사운드 엔진을 제작하였다.

### [Utility]

장면(Scene), loading, saving

Detailed debug and performance reporting

dump 분석 및 비정상 에러 검출을 위한 파일 전송 엔진 시스템 구축.

Resource Manager

그래픽 리소스 및 기타 리소스 관리를 위한 시스템 구축

Efficient Memory Manager

고가의 바운스 체커를 제외하고라도 실시간으로 메모리가 중첩되거나 혹은 잘못 쓰여진 경우의 에러 검출 시스템 등이 필요하였다. 메모리의 중첩, 누수 등을 위한 가벼운 자체 시스템을 제작하였다.

KeyBoard, Mouse And JoyStick Input

사용자 입력 처리와 아케이드 느낌을 살리기 위한 joystick을 지원하기 위한 Utility를 제작하였다.

### 충돌 검출

Point, line, area – based object picking 지원

Area – area 충돌 검출 지원 ex) 원과 사각형, 원과 원 등의 야구 특성과 신 야구의 타격 시스템에 맞는 충돌 검출을 제작하였다.

네트워크 엔진

서버와의 통신을 담당하는 부분과 클라이언트간 통신을 담당하는 부분 양쪽 모두의 처리가 가능하도록 작업을 진행하여야 되었다. 신 야구의 네트워크 엔진은 위의 모두를 만족하여야 하였으며, 패킷 암호화를 통하여 패킷을 보호하도록 제작하였다.

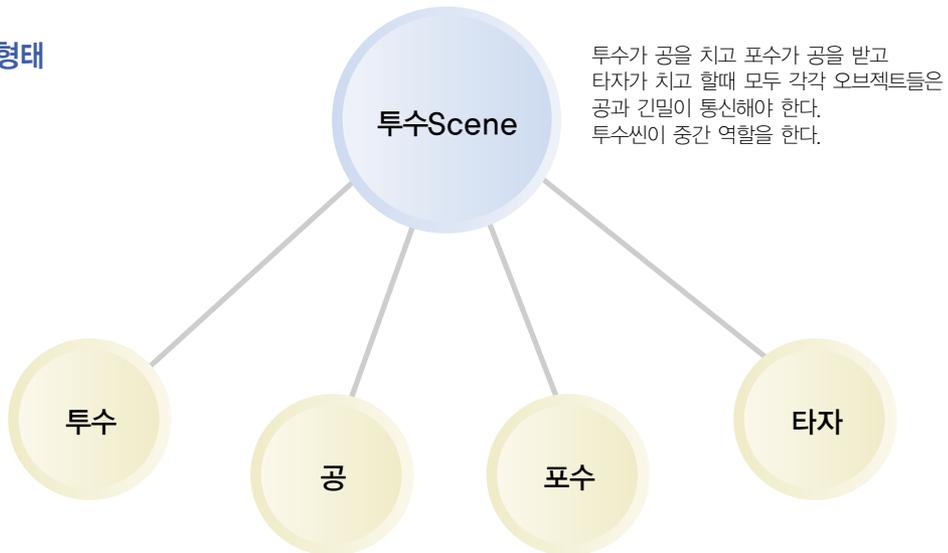
〈신야구의 각 씬에 대한 설명〉

신야구의 각각의 씬 구성은 크게

1. 투수가 타자와 포수를 바라 보는 위치에서의 씬
2. 타자가 투수와 센터라인을 바라보는 씬
3. 공을 기준으로 조감도의 위치에서의 씬

이렇게 세가지로 나뉜다. 각각의 구성요소를 보면,

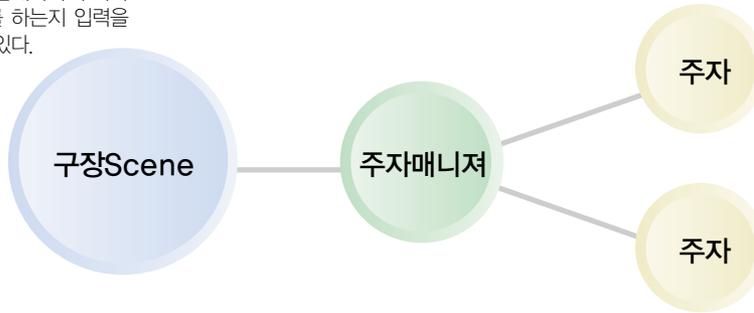
투수뷰 형태



씬그래프의 한 단계인 투수 씬의 구성이다, 각각의 야수와 야수에서 특화된 객체인 포수 , 타석에 들어 서 있는 타자 등이 렌더링할 타겟으로 하여 한 개의 씬으로 구성된다.

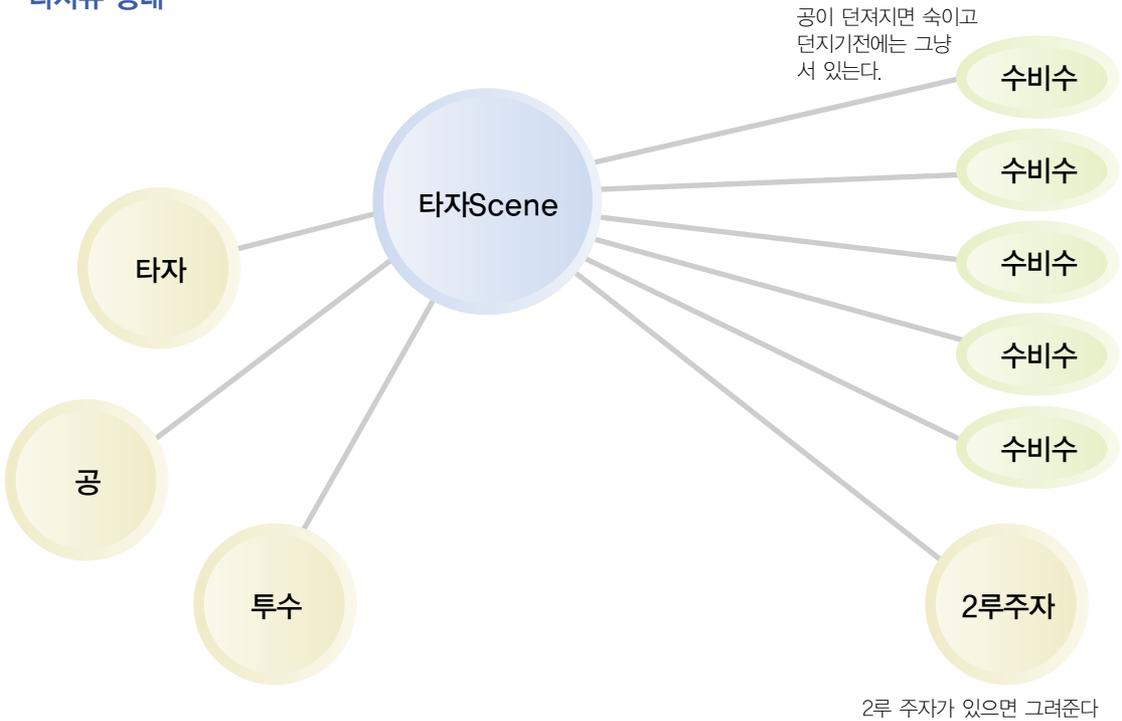
구장씬과 주자의 관계이다.

구장씬은 메모리상에 계속 존재하며 주자가 도루를 하는지 입력을 받고 있다.



### [타자 씬의 모습]

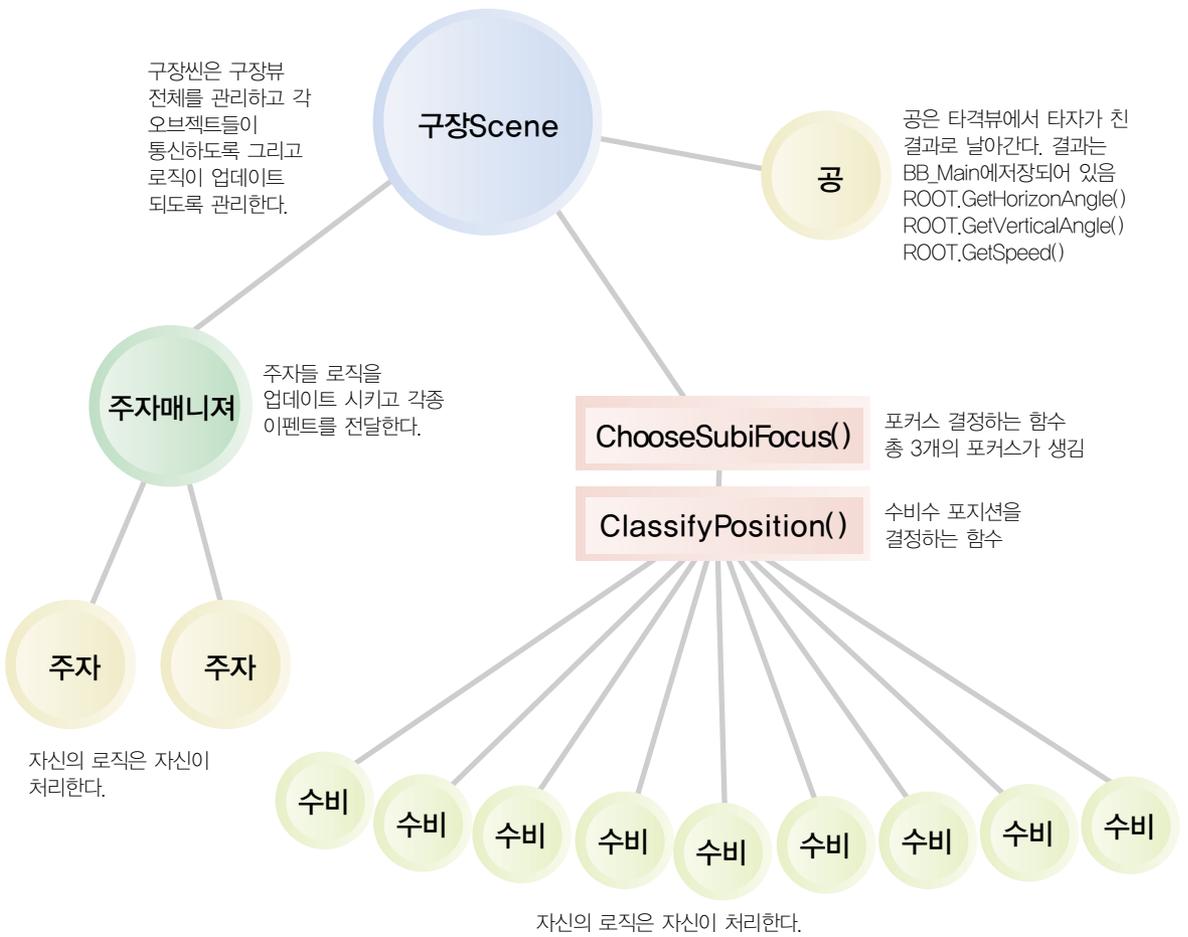
#### 타자뷰 형태



씬구성 요소 중 하나인 타자 씬에 대한 설명이며 타자의 시점에서 투수 및 센터라인을 바라보는 구성 요소이며 공격하는 타자와 투수 각각의 수비수와 2루 주자를 렌더링하기 위한 각각의 노드들로 구성되어 있다

### [구장 씬의 모습]

#### 구장뷰 형태



조감도와 같은 모습의 씬을 구성하는 구장 씬에 대한 설명이다.

위의 내용과 같은 씬으로 구성되며, 주자와 수비 등이 렌더링을 위한 노드를 가진다.