캐릭터 이동과 카메라 처리

부 서 명	개발6팀	작 성 자	이희성
최초작성일	2009-05-13	최종수정일	2009-05-13

작성 일지(업데이트 기록)

일시	내용	작성자

목차

1. 캐릭터 이동과 카메라 처리

1.1. 카메라

1.1.1. 위치

- 기본 위치 값 : 캐릭터 좌표(X, Y+1, Z+2.3)
- 높이값 범위(Z축 이동 범위값) : 2m ~ 6m
- 거리값 범위(Y축 이동 범위값) : 0m ~ 15m

1.1.2. 회전&줌(I/O) 속도

- 회전 : R-Speed(angles per second)는 옵션으로 조정가능, Default 15
- 줌(I/O) : Wheel Speed는 옵션으로 조정가능, Default 30

1.1.3. 카메라 설정

T-Speed(meters per second), Wheel Speed

- 카메라 B 는 아래의 그림 2 와 같이 캐릭터의 머리 정중앙 A(그림 1 참조)를 주시한다.
- 화면 확대/축소에 따라 카메라는 카메라 이동 경로 Z 를 따라 포물선 이동한다.

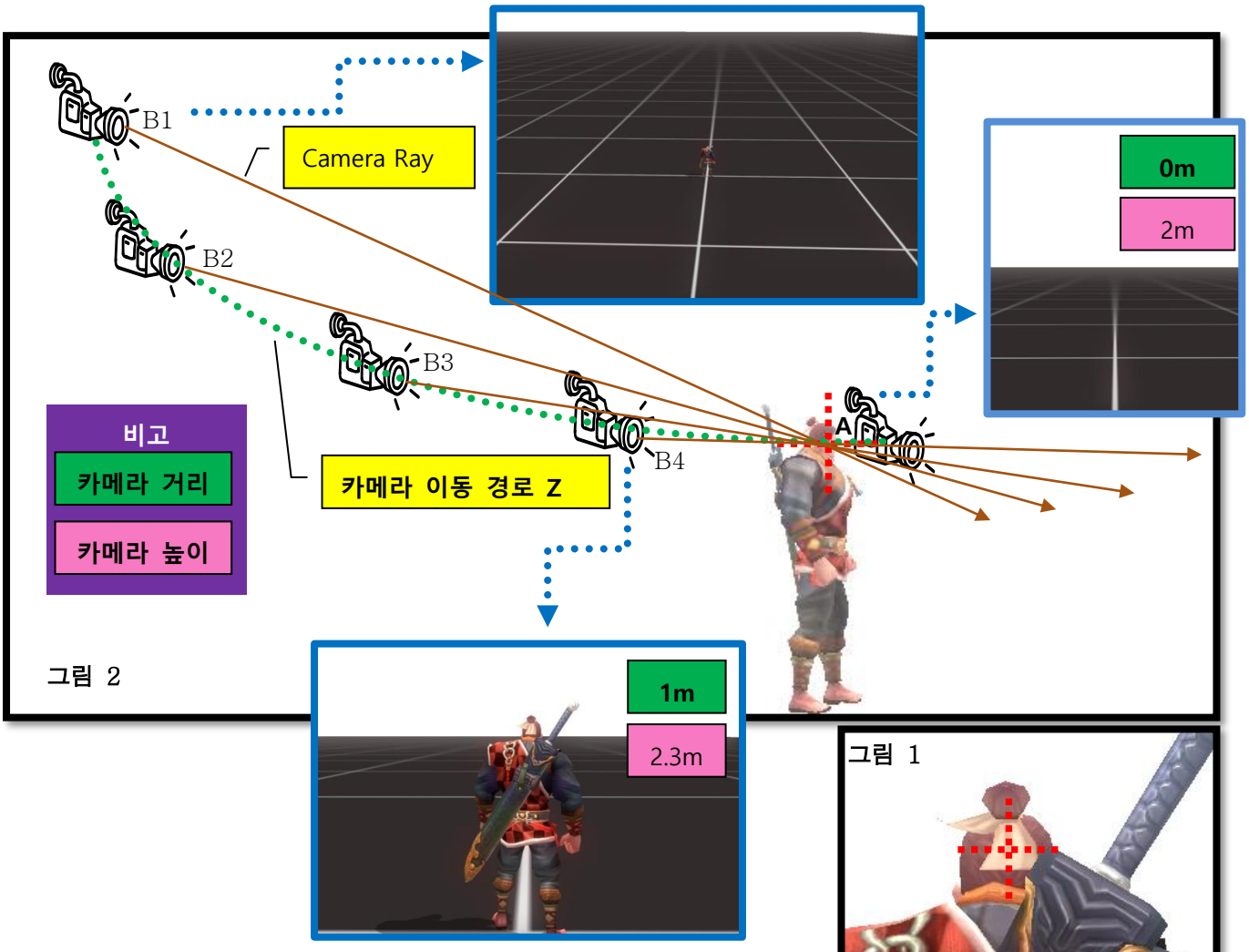
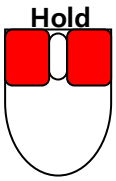


그림 2

그림 1

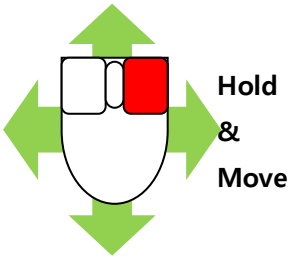
1.1.4. 카메라 조작

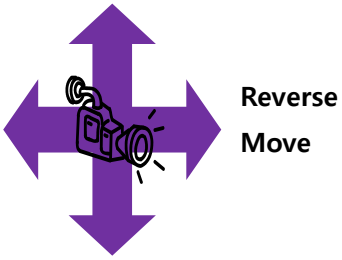
- 마우스 오른쪽 OR 왼쪽 버튼(Mouse R_Button)을 Hold한 시점에서 카메라뷰를 인위적으로 조종이 가능하다. R_Button Hold는 카메라의 방향에 캐릭터의 방향을 맞춰주며, L_Button Hold는 캐릭터의 방향을 이전상태로 유지 해주는 차이점을 가진다.



***이때 커서 포인트는 해제시킨다.**
(화면에서 사라짐)

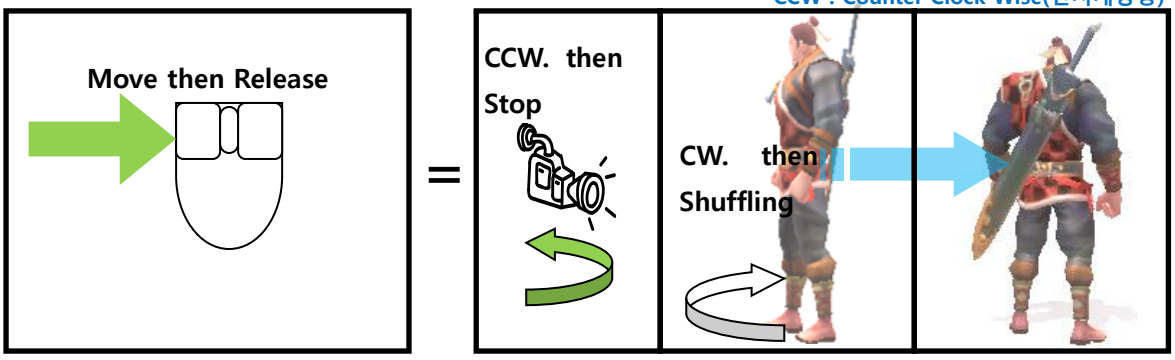
- 즉, R_Button Hold한 후 마우스를 이동하면 이동하는 반대 방향으로 카메라의 방향이 변경된다.





- 이때, 카메라의 수직 각도는 상향 0°~90° 하향 0°~270°까지 범위에서 이동가능하며,
- 수평 각도는 좌우측 90°까지는 카메라만 이동(캐릭터는 상하체 분리에 따라 시선은 카메라 방향 주시, 하체는 고정)하며, 90° 이상이 넘어가면 5°증가시 해당 도수 만큼 좌우측 Shuffling이 발생한다.
- R_Button이 Release되면 카메라는 해당 방향에 고정되며, 캐릭터 방향이 카메라 방향에 맞춰지면서 Shuffling이 발생한다.

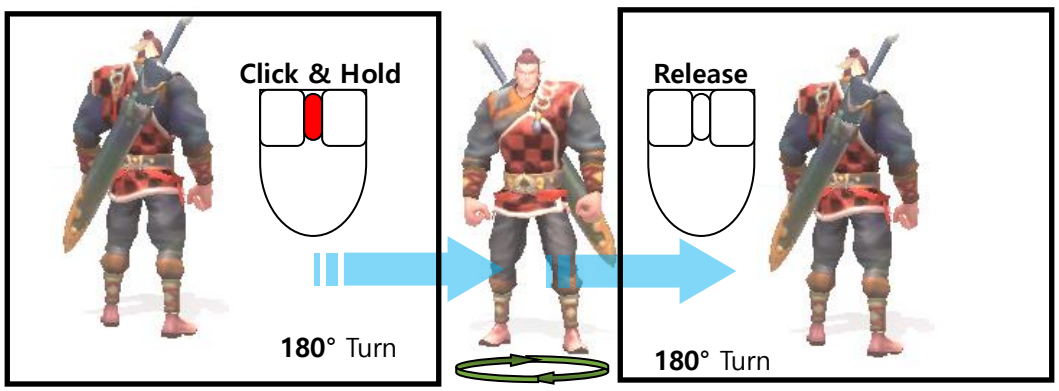
*CW : Clock Wise(시계방향)
*CCW : Counter Clock Wise(반시계방향)



- 카메라의 이동 속도, 회전 속도, 줌(I/O)속도는 옵션창의 눈금단위에 비례하여 증감한다.

1.1.5. 부가 기능

- 화면반전 : Wheel Button Click & Hold시 카메라 CW방향으로 180°회전 지속, Release시 해제



1.2. 이동

- 무극의 모든 캐릭터는 NaviMesh로 설정된 테이터 구조물위를 이동한다.
- NaviMesh는 맵툴에서 설정한다.
- NaviField의 Field Value의 값이 1로 설정된 NaviMesh로는 이동이 불가능하다.

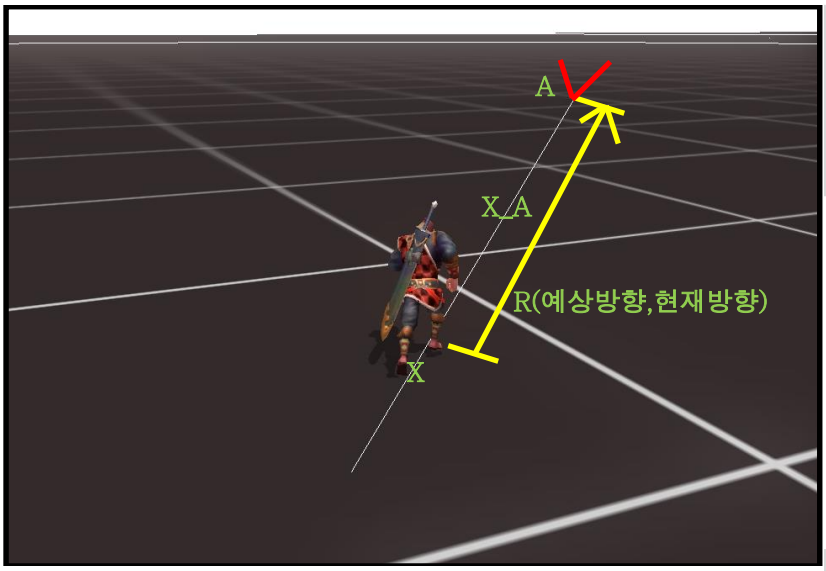
1.2.1. 키보드 이동조작에 따른 카메라 처리

- 키입력에 따른 카메라 처리

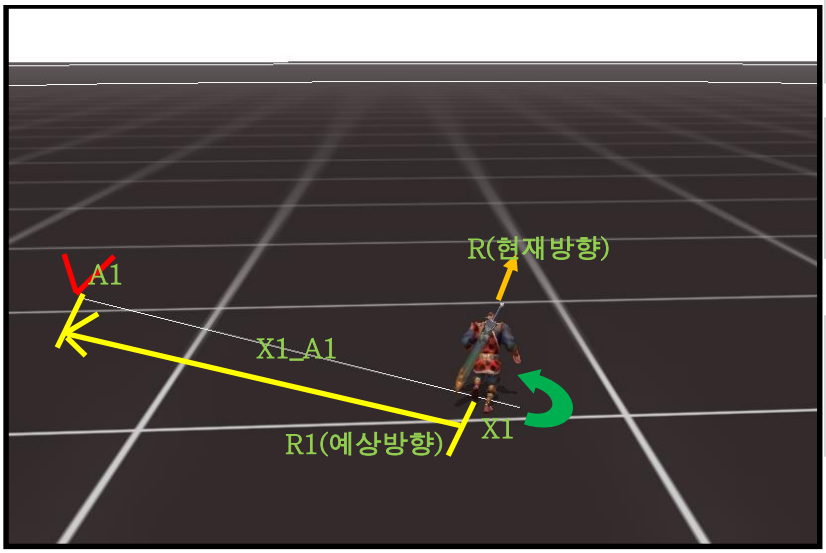
키 OR 마우스입력	캐릭터 처리	출력 애니메이션	카메라 처리
/	걷기 ↔ 뛰기 토글	해당항목	~
W	캐릭터 전방이동	ForwardStep	전방이동
S	캐릭터 후방이동	BackwardStep	후방이동
A	캐릭터 좌 회전	LeftShuffle	반시계방향 회전
D	캐릭터 우 회전	RightShuffle	시계방향 회전
Q	캐릭터 좌측 이동	LeftStep	좌측 이동
E	캐릭터 우측 이동	RightStep	우측 이동
W+A	캐릭터 좌측 호선 이동	ObliqueLeftStep	좌측 호선 이동
W+A+Mouse R_Button Hold	W+Q 입력과 동일 (캐릭터 좌측 사선 이동)		
W+D	캐릭터 우측 호선 이동	ObliqueRightStep	우측 호선 이동
W+D+Mouse R_Button Hold	W+E 입력과 동일 (캐릭터 우측 사선 이동)		
S+A	캐릭터 후방 우측 호선 이동	ObliqueRightBackwardStep	후방 우측 호선 이동
S+D	캐릭터 후방 좌측 호선 이동	ObliqueLeftBackwardStep	후방 좌측 호선 이동
W+Q	캐릭터 좌측 사선 이동	ObliqueLeftStep	좌측 사선 이동
W+E	캐릭터 우측 사선 이동	ObliqueRightStep	우측 사선 이동
S+Q	캐릭터 후방 좌측 사선 이동	ObliqueLeftBackwardStep	후방 좌측 사선 이동
S+E	캐릭터 후방 우측 사선 이동	ObliqueRightBackwardStep	후방 우측 사선 이동
Mouse R_Button Hold	없음	유지	포커스 Hold
A+Mouse L_Button Hold	캐릭터 좌 회전	LeftShuffle	현재 상태 고정
D+Mouse L_Button Hold	캐릭터 우 회전	RightShuffle	현재 상태 고정
A+Mouse R_Button Hold	Q 입력과 동일		
D+Mouse R_Button Hold	E 입력과 동일		
Mouse L_Button Hold & Move	유지	유지	마우스 이동 방향
Mouse R_Button Hold & Move	카메라 방향=캐릭터 방향	Case By Case	마우스 이동 방향
Mouse M_Button Wheel Up	유지	유지	줌인
Mouse M_Button Wheel Down	유지	유지	줌아웃
Space Bar(정지/걷기 상태)	점프	JumpStart,Loop,End	유지
Space Bar(달리기 상태)	점프	JumpStart,JumpLoop,JumpRun	유지
Mouse L_Button & R_Button Hold	W 입력과 동일		

1.2.2. 마우스 이동

① NaviMesh상에 Mouse Curser가 위치되고 Click이 이루어지는 상황을 지면포인팅이라 한다.
   A. 지면상에 포인팅이 이루어지면, 지면포인팅 이펙트가 출력된다.
   B. 지면포인팅 이펙트는 별도로 제작되어 등록되며, 1~2초 이내의 짧은 이펙트로 처리한다.
② 캐릭터는 캐릭터의 현재위치 X,Y,Z에서 지면포인팅된 X1,Y1,Z1으로 직선 이동한다.
③ 모든 위치좌표 X,Y,Z는 NaviMesh상에 위치해야 한다.
④ 이동 중에 이동불가능 상황이 발생하면 A*(A-Star)를 이용한 길찾기를 시행한다.
   A. Step-In 높이 이상의 경우
   B. NaviField값 1의 경우
   C. Swim-In 의 경우
⑤ A* 길 찾기가 더 이상 실행되지 않는 상황인 경우 해당 위치에서 정지한다.
⑥ 캐릭터의 현재 방향에서 마우스 포인팅된 지점까지의 일직선상의 방향으로 캐릭터 방향 전환이 이루어진다.



지면A를 포인팅함에 따라 현재 좌표 X와 점 A를 잇는 일직선상의 X_A를 이동하게 된다.



X_A를 이동중 지면A1에 다시 포인팅

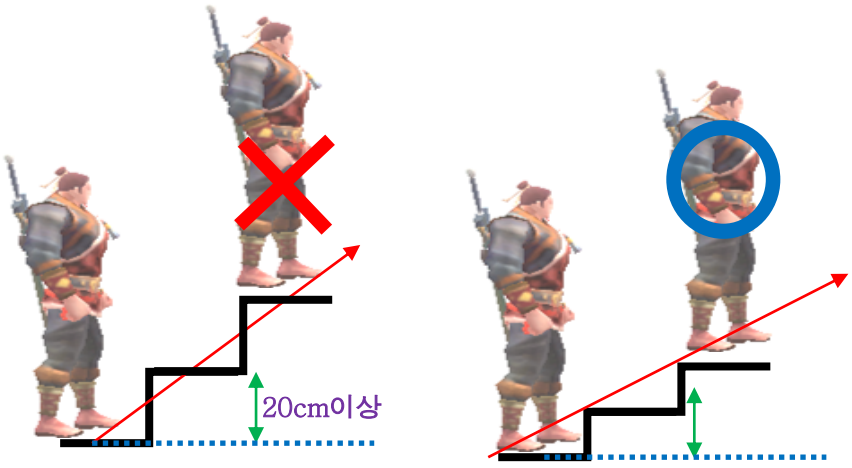
X1위치,현재방향R에서 예상방향 R1으로 CCW캐릭터 회전

X1위치와 A1을 잇는 직선 X1_A1을 이동

⑦ 점프키의 입력에 의해 포인팅 지점이 초기화 되지 않는다.
⑧ 키보드 이동에 사용되는 키 입력이 진행되는 경우 즉시, 해당 키가 적용된다.
⑨ 마우스 이동은 게임옵션창에서 On/Off가 가능하며, Default는 Off로 설정된다.
⑩ 카메라 조작키는 키보드 이동 조작키와 동일하다.
⑪ 마우스 포인팅 키의 입력값은 클라이언트상에서는 윈도우 제어판의 마우스 등록정보 값만큼 출력되지만, 게임서버로의 패킷전송은 2times per sec.(추후 재 조정 가능) 단위로 갱신한다.
   A. 오토마우스 등에 의해 전송되는 과다한 패킷전송을 서버측에서 필터링 하기 위함.
   B. 대규모 전쟁시에 서버 부하를 감소시키기 위한 논의 필요.

1.2.3. Step-In (미적용 상태, 적용 논의필요)

① 캐릭터 이동시 별도의 조작(점프) 없이 진입할 수 있는 높이값을 Step-In이라 명칭한다.
   A. 무극에서 가장 작은 캐릭터인 동영여자 캐릭터(신장 80cm)를 기준으로 삼는다.
   B. 80cm의 1/4인 20cm를 Step-In높이로 설정한다.(무릎위)
   C. 즉, 20cm이하의 높이차를 가지는 지면은 별도의 조작 없이 이동키,마우스 피킹을 이용하여 이동이 가능하다.
   D. 20cm이상의 높이 차를 가지는 지면은 점프에 의해 이동이 가능하며, 점프 높이값을 상회하는 고저차이는 진입이 불가능하다.



   E. 또한, 20cm이하의 높이값을 가지지만, 60°이상의 경사도는 이동이 불가능하다.

1.3. 충돌 체크

1.3.1. 카메라 충돌 체크

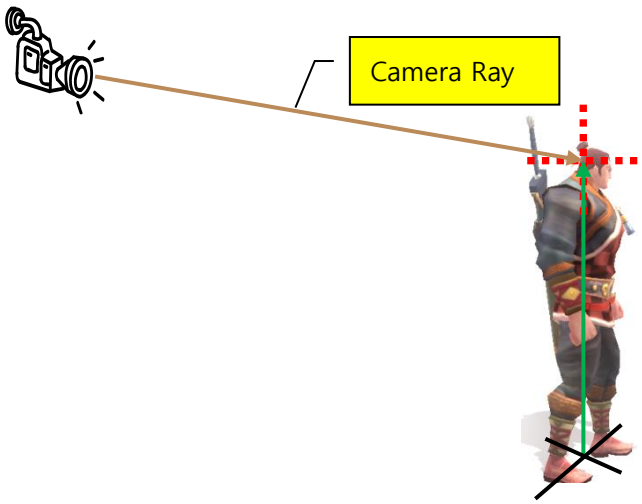
- 캐릭터의 시야권을 제공하기 위해 자동으로 조정되는 카메라 시스템을 위해 카메라 충돌체크를 검사한다.
- 즉, 캐릭터와 카메라 사이에 시야를 방해하는 객체가 존재할 경우 인위적으로 카메라를 개별 조정 하지 않아도 해당 객체를 투영하거나, 해당 객체 앞으로 카메라를 자동적으로 이동 시키기 위함이다.

① 캐릭터 조준점

A. 종족별,성별로 분류된 PC캐릭터의 머리 뒤통수 중앙점으로 변경
(현재, 캐릭터 중심점으로 설정되어있음)

B. 해당 캐릭터의 위치좌표에서 일정 고정값들을 개별 가산하여 해당 조준점을 계산한다.
(북방남,북방여,남방남,남방여,중원남,중원여 캐릭터의 높이값이 모두 각각이기 때문)

계산된 조준점은 그림과 같이 캐릭터의 머리 뒤통수 중앙에 위치됨.



캐릭터 위치좌표점에서 수직방향으로 캐릭터 마다의 일정 고정값들을 개별 가산하여 조준점을 도출.

② 카메라 레이(Camera Ray)

A. 카메라에서 캐릭터 조준점 까지의 가상의 Line

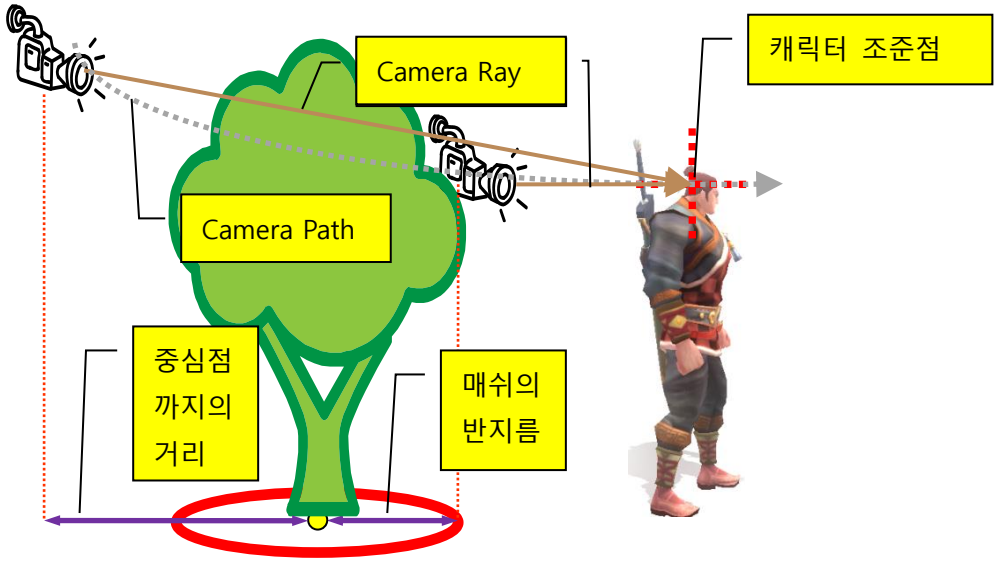
B. 카메라 충돌 체크에 사용되며, 별도의 조작이 없는 경우, 유저가 자신의 캐릭터를 바라보는 방향이 됨

C. Mouse M_Button의 Wheel Up/Down이 이루어짐에 따라 카메라 레이의 길이는 확대/축소 됨.

③ 충돌 처리

A. Camera Ray에 오브젝트 매쉬가 겹쳐지는 순간을 카메라 충돌 이라 한다.

B. 카메라 충돌이 발생하면, Camera Ray와 가장 가까운 매쉬 정보를 검색한후 해당 매쉬의 중심까지의 거리 + 해당 매쉬의 반지름 만큼을 카메라 줌인하여, 캐릭터를 가리는 매쉬를 시야에서 처리한다.



④ 예외사항

A. 카메라 충돌처리는 지형과 지형오브젝트에 대해서만 적용된다.

i. 캐릭터(NPC, PC, Monster)는 카메라 충돌 처리 하지 않는다.

B. 지형 오브젝트들 중 일부분에 대해서는 아래의 방법으로 처리한다.

i. 카메라 충돌 처리가 필요하지 않은 특정 오브젝트는 Flag 정보를 삽입하여, 이를 통해 카메라 충돌 처리의 여부를 판별 한다.

1.3.2. 캐릭터 충돌 체크

- 무극온라인은 캐릭터의 충돌 체크를 사용하지 않는다.