FPS 사격장 (개인 프로젝트)

작성자 : 홍진호
작성일 : 2018-05-18

목차

- 개발 정보
- 총돌리기 구현.
- 케릭터 이동 구현.
- 총알 발사

개발 정보

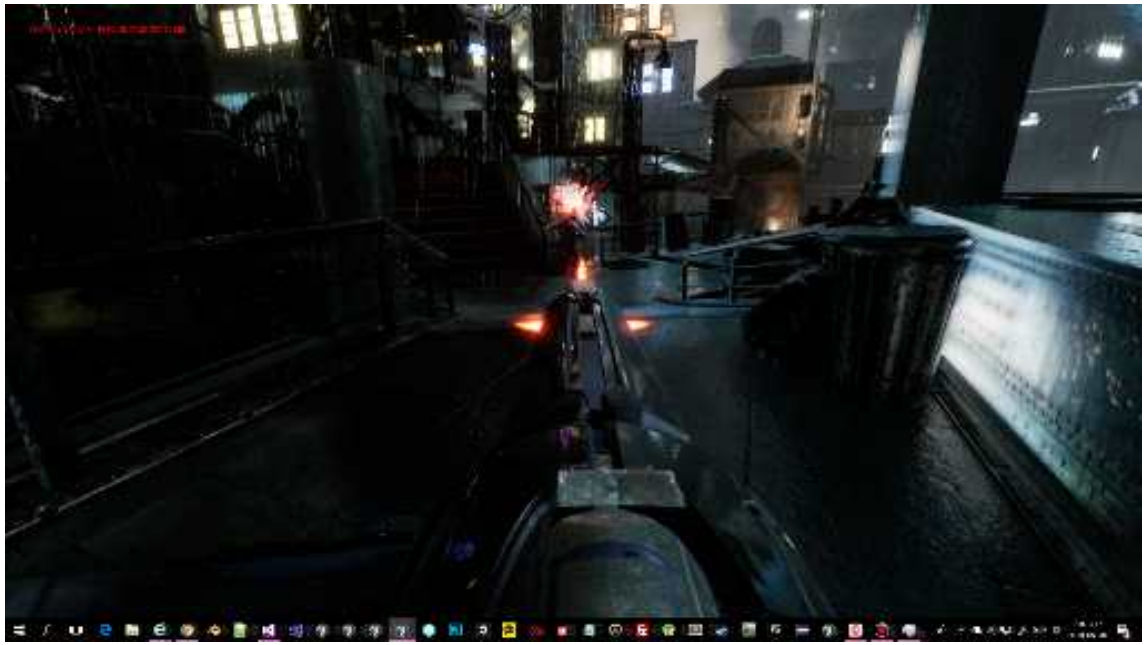
- 개발 툴 : 언리얼엔진 4.19.2
- 언어 : c++ , 블루프린트(스크립트)
- 리소스 출처:
  - 3d 리소스, 애니메이션 : 마켓 플레이스 언리얼 쇼케이스 Paragon Twinblast , Paragon Murduck
  - 총소리 : http://soundbible.com/2091-MP5-SMG-9mm.html
- 풀소스 경로 :https://drive.google.com/open?id=1P1XFwFGi2awve1Em0mX_UIdXWUOEwnlr
- 실행 파일: https://drive.google.com/open?id=1X6IONr35XI5vcVwoZWbU6KXQsg6YwoAZ
- 플레이 영상 : https://youtu.be/SSt_0se8XEU
- 관련 함수들은 소스 폴더에 모아놓았습니다.

총돌리기 구현.

FPS에서 가장 기본적인 기능입니다.

인풋 옵션에서 mouse_turn,
mouse_up_down이라고
이름 지은 축이벤트를 추가하여
SetupPlayerInputComponent 에 바인딩
시켜 마우스의 움직임을 감지합니다.

로직의 핵심 함수는
move_front_and_back , move_left_right
함수입니다.



총 돌리기 수평 구현.

- 함수에서는 마우스 수평 회전값를 받아옵니다.
- 마우스 수평 회전 값이 0일 경우 작동하지 않습니다.
- 마우스 수평 감도와 틱 값을 곱셈하여 프레임 대응 회전 값을 얻습니다.
- 위에서 구한 값에 마우스 수평 회전 값을 곱하여 회전 변화율을 얻어서 ControlRotation에 yaw 축으로 회전 값을 더해줘서 마우스 수평 회전 기능을 구현합니다.

총 돌리기 수직 구현.

- mouse_up_down 함수는 기본 로직은 mouse_turn 함수와 거의 유사합니다.

- 단 mouse_turn 함수는 controller가 타켓이지만 수직 마우스 회전에는 캐릭터 스켈레탈 메시의 총구에 소켓을 붙여서 애니메이션(블랜드 스페이스)의 변수를 제어하여 카메라가 애니메이션을 따르게 하였습니다.

- 카메라가 상하로 너무 회전하였더니 어지럼증을 유발하여
  카메라의 피치 값을 ±50으로 제한을 걸었습니다.

캐릭터 이동 구현.

- 인풋 옵션에서 front_back과 left_right이라고 이름을 지어서 축 이벤트를 추가합니다.
- front_back 축 이벤트 -> move_front_and_back 함수 실행
- left_right 축 이벤트 -> move_left_right 함수 실행

- 두 함수 다 로직은 같습니다.

- 캐릭터의 방향 벡터를 얻어와서 movement에게 축 값과 방향 벡터를 추가하여 캐릭터를 이동시킵니다.

- 직진이나 후진을 할 때는 달리는 모션을 제어하는 I_RUN_now 변수를 0.4로 하여 총 돌리기 모션과 알파 블렌딩을 하여 총 돌리기 모션과 같이 적용하게 했습니다.

총알 발사.

- 0.4 초 동안 5발의 총알이 나가는 총을 개발.

- 주요 로직 : 인풋의 축이벤트를 사용하여 클릭 시 0초에 한발 0.1초에 한발, 0.2초에 한발, 0.3초에 한발, 0.4초에 한발을 스폰하여 0.4초가 지나면 attack_locker를 해제하여 다음 발사가 작동합니다(발사는 타이머 핸들러로 병렬처리를 수행합니다).

  발사 타이머를 받는 변수를 추상적 자료형이 아니고 배열을 쓴 이유는 타이머가 총 4번 정해진 개수만큼 수행하기 때문에 고정크기를 가지는 배열을 썼습니다.

- 총알 스폰 위치 스켈레탈 메시에 달아놓은 소켓의 transform을 얻어와 회전 값과 위치 값을 얻어와 총알에 transform을 적용 시키면서 스폰합니다.

  생성한 총알 액터에서 만든 shot 함수를 호출하여 발사 후 총알의 물리 처리 같은 기능을 수행할 수 있게 하였습니다.

총알 액터

- 총알이 발사되면 그 순간 설정된 초기속도를 기준으로 수평 방향으로 이동하는 물리 식이 계산되고 z축은 중력가속도를 적용한 속도 값을 얻습니다.

- z 중력가속도 적용(등속가속도 운동) : 속도(변위) = 속도(z) + (중력가속도* 시간)

- 수평 이동 변위 = 수평 속도 변위 값 * 시간 변위(DeltaTime)
- 수직 이동 변위 = 수직 속도 변위 값 * 시간 변위(DeltaTime)

- 위에서 계산한 값들을 SetActorLocation(원래 위치 + 이동 변위)를 사용하여 총알을 이동시킵니다.

총알 액터

- SetActorLocation 을 사용하면 overlap 이벤트 인식률이 떨어집니다.
- 시작 위치에서 -> 최종 이동한 위치로 바로 이동하기 때문입니다.
- 이를 해결하기 위해서 ray cast를 사용하여 시작위치에서 최종 이동한 위치까지 추적하여 오버랩에 pawn 클래스가 걸리면 pawn 클래스의 OnComponentBeginOverlap.Broadcast 를 호출하여 오버랩 이벤트를 발생시킵니다.
- player_char 클래스의 오버랩 이벤트 피격을 인식하여 체력이 까지는 기능과 피격 이펙트를 생성하고 피격한 총알을 제거합니다.

  체력이 0이 되면 사망 애니메이션을 실행합니다.