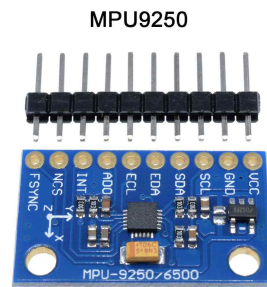


Balance Ball (Serial Test & Shooting Range)

- Balance Ball 콘텐츠(사격), 시리얼 통신
- Unity3D로 구현
- 하드웨어, 시리얼 통신

1. MPU9250



- 9축 자이로, 가속도, 지자기 센서
- 자이로, 가속도만 사용하였음

2. Balance Ball



(a) 거치대, 핸들



(b) 밸런스 볼



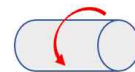
(c) 결합 후

3. 센서 각도 계산

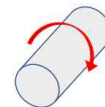
센서 각도 계산

● 가속도

• X축 회전량 : $angle(X) = \tan^{-1}\left(\frac{AcY}{\sqrt{Ac^2X + Ac^2Z}}\right) \times \left(\frac{180}{\pi}\right)$



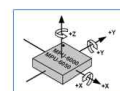
• Y축 회전량 : $angle(Y) = \tan^{-1}\left(\frac{-AcX}{\sqrt{Ac^2Y + Ac^2Z}}\right) \times \left(\frac{180}{\pi}\right)$



- Z축 회전량 : 중력 가속도와 정반대의 방향 -> 구할 수 없음

● 자이로

• 각속도 적분 : $\theta = \int_0^{t_f} \omega(t) dt$



- 하드웨어

4. 상보 필터 사용

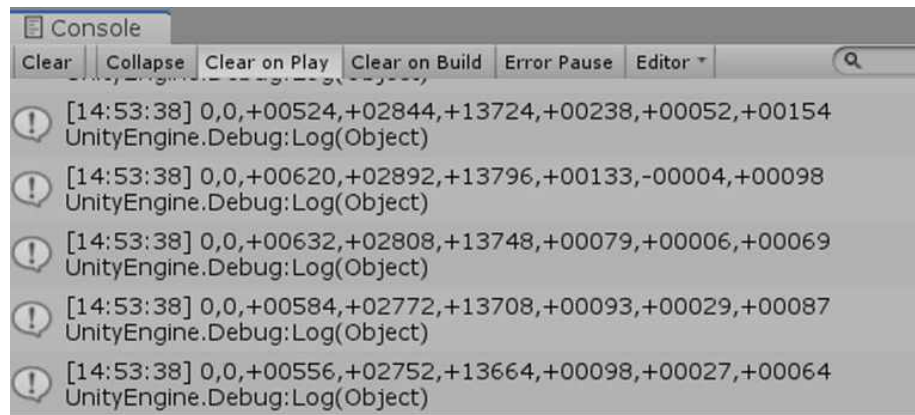
상보 필터

$$\text{Filtered Angle} = \alpha \times (\text{Gyroscope Angle}) + (1 - \alpha) \times (\text{Accelerometer Angle})$$

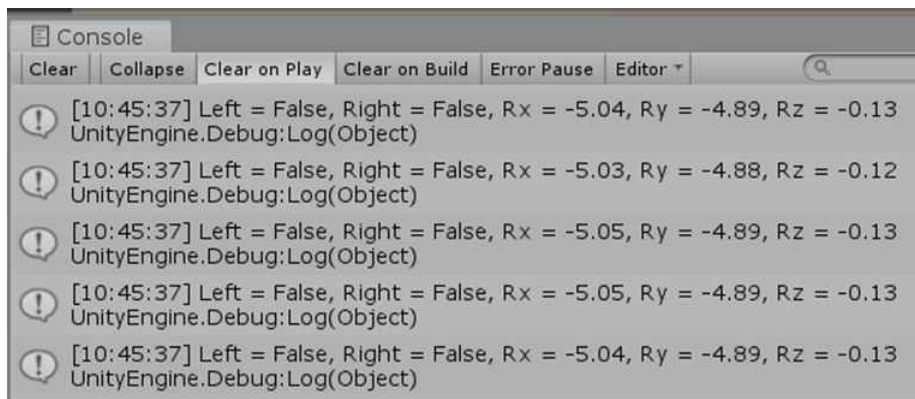
- 가속도 센서
 - 오차 누적 X
 - 노이즈, 이동, 진동에 취약함
 - Z축 회전량을 알 수 없음 (자이로 센서로 보충 가능)
- 자이로 센서
 - 적분 시, 오차 누적 현상(드리프트) 발생

5. 가공, 보정

- 초기값



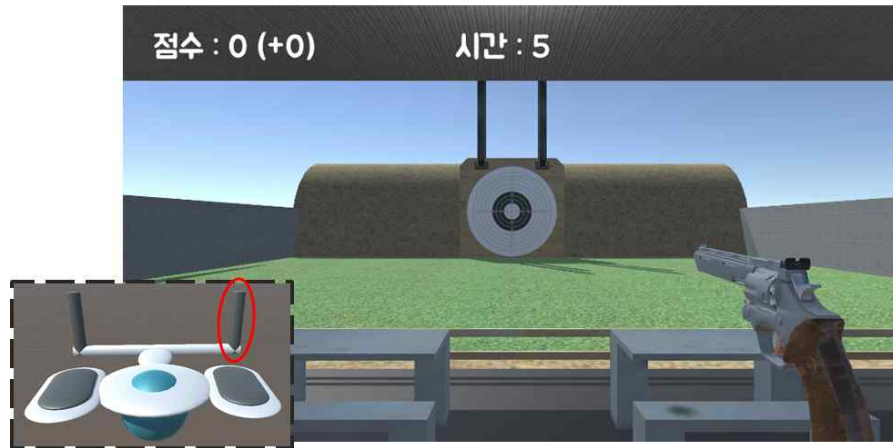
- 수정값



- 화면구성(사격)

- 현재 마우스, 키보드를 사용하여 구현되었음 (BalanceBall 연동 필요)

1. 메인화면

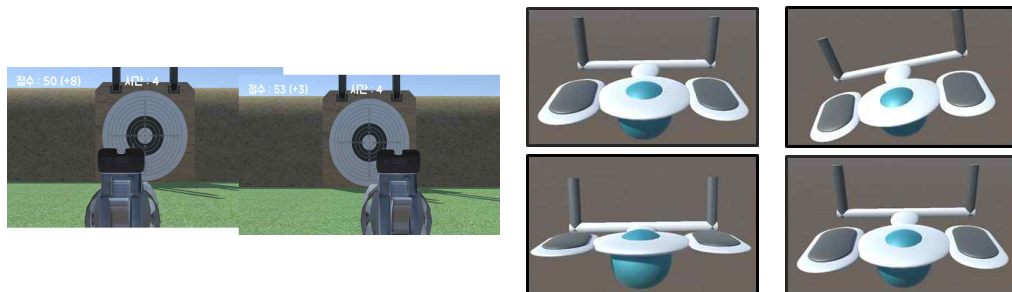


- 화면 전환(ZOOM) : Balance Ball 오른쪽 버튼 클릭
- 현재 : 마우스 오른쪽 버튼 클릭

2. 게임 진행



- ZOOM 화면
- Zoom In 상태 시, 시간 카운트 감소
- 한 번 더, 오른쪽 버튼을 누르면 Zoom 상태가 풀림



- 화면 확대 시, 총이 랜덤 방향으로 이동
- Balance Ball 조절로 총 이동 방향 조절 가능
- 현재 : 키보드 ←↑→↓로 조절할 수 있게 구현됨

- 화면구성




- Balance Ball 왼쪽 버튼 클릭 시, 총 발사
- 발사 시, 정확도에 따라 점수 표시(1점~10점)
- 현재 : 마우스 왼쪽 클릭

- Assets (Serial Test)

1. Scene 구성


이름

 SampleScene.unity

- 총 1개의 Scene으로 구성 - SampleScene

2. Scripts 구성 (사용하는 주요 스크립트)

이름


 SerialTest.cs

- SerialTest.cs
 - raw data 가공, 보정 (현재는 하드웨어팀에게 넘겨 수정 완료 상태)
 - data 받아오기 (이름 통해 Unity 내 Object, Event와 연동할 것)

- Assets (Shooting Range)

1. Scene 구성


이름

 SampleScene.unity

- 총 1개의 Scene으로 구성 - SampleScene

2. Scripts 구성 (사용하는 주요 스크립트)

이름

 GameManager.cs

- GameManager.cs - 게임 이벤트 관리 스크립트