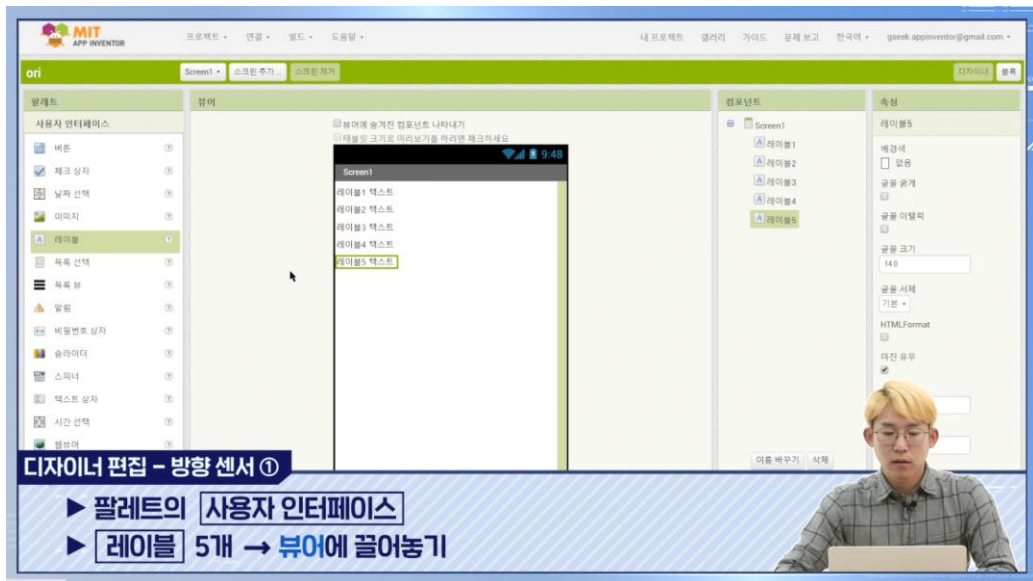


# 8. 방향 센서를 활용한 공굴리기 앱

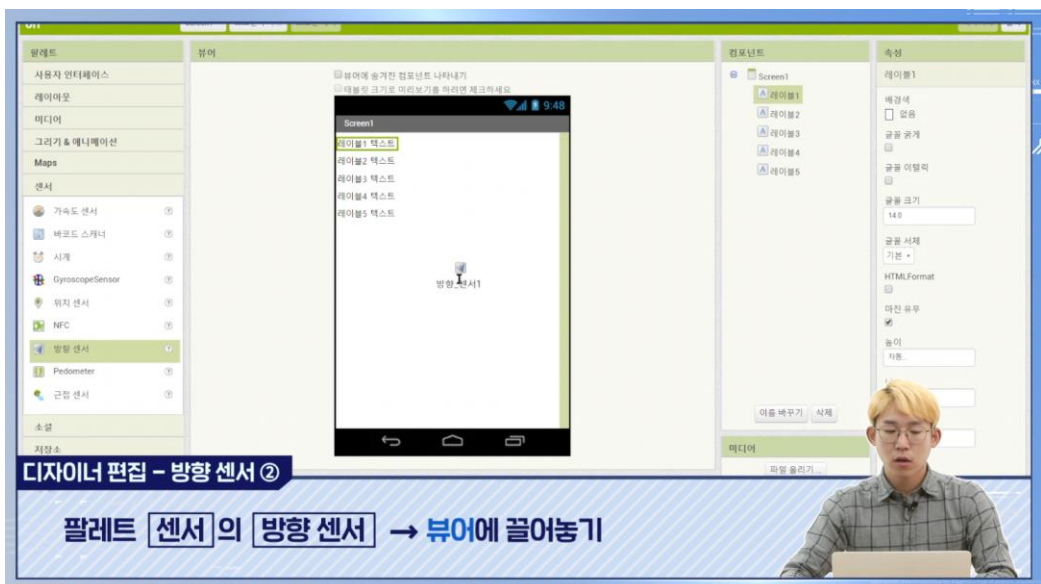
## 1. 방향 센서를 활용한 공굴리기 앱 만들기

### 1) 디자이너 편집

① 팔레트 [사용자 인터페이스]의 [버튼] → 뷰어에 끌어놓기



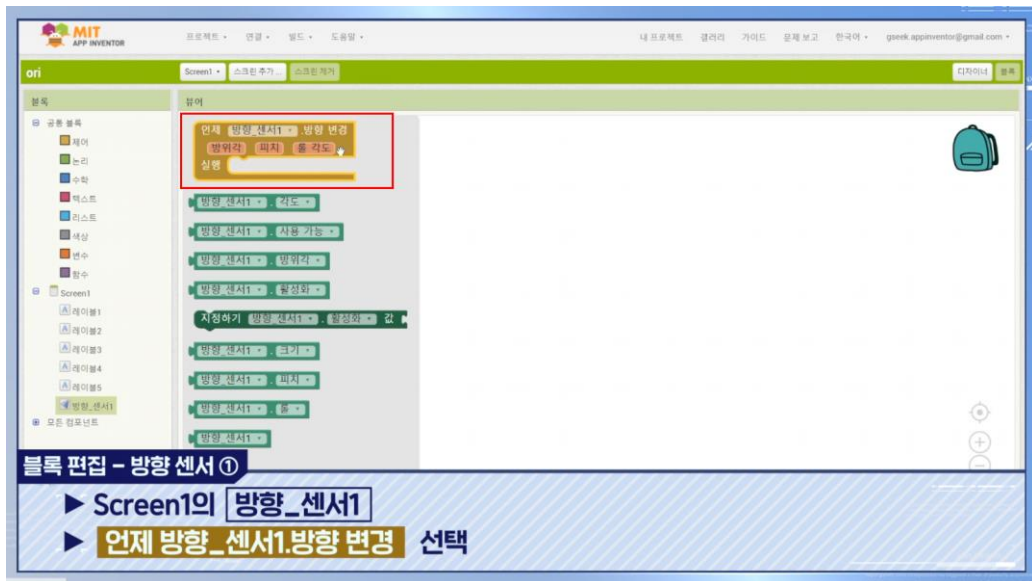
② 팔레트 [센서]의 [방향 센서] → 뷰어에 끌어놓기



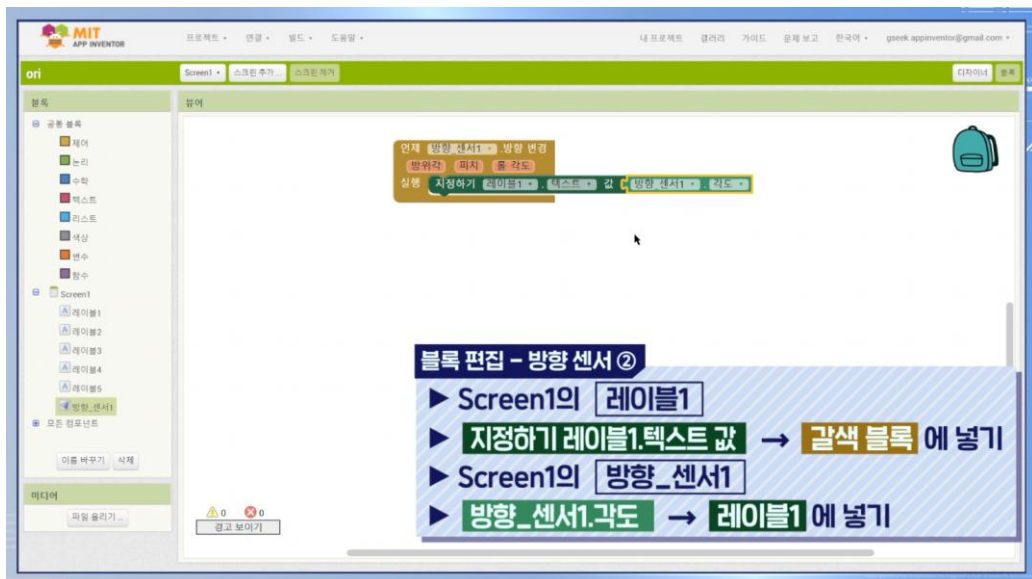
# 8. 방향 센서를 활용한 공굴리기 앱

## 2) 블록 편집 - 방향 센서

① [screen1]의 [방향\_센서], [언제 버튼\_센서1.방향 변경] 선택

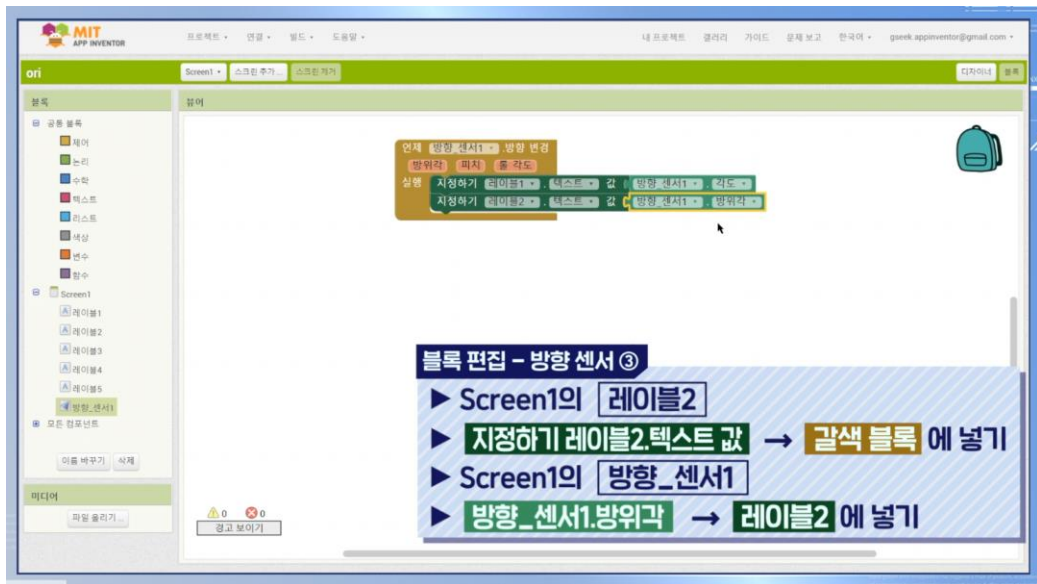


② [screen1]의 [레이블1], [지정하기 레이블 텍스트 값]  
 → [갈색 블록]에 넣기, [screen1]의 [방향\_센서1] [방향\_센서1.각도]  
 → [레이블1]에 넣기

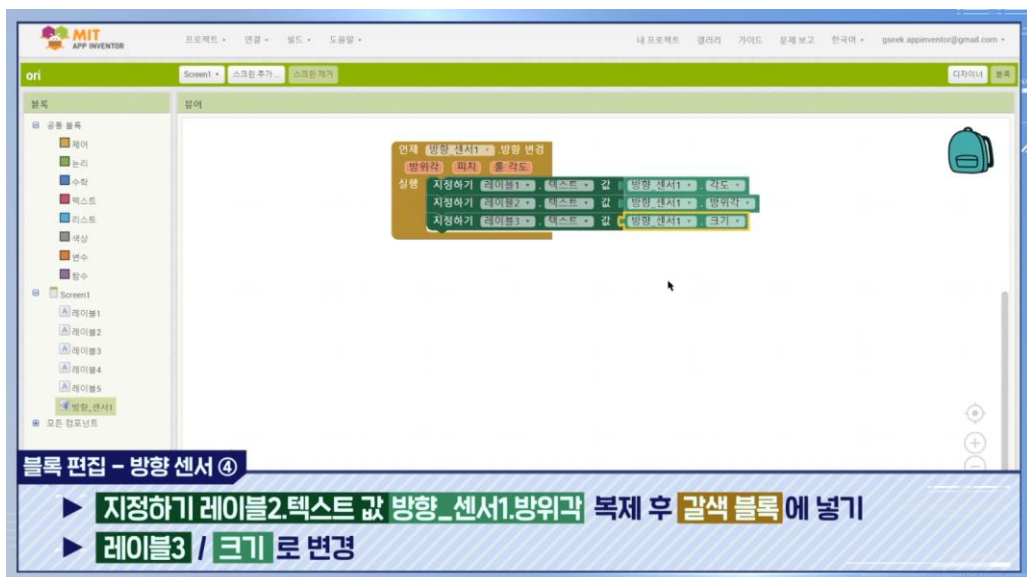


## 8. 방향 센서를 활용한 공굴리기 앱

- ③ [Screen1]의 레이블2, [지정하기 레이블2.텍스트 값]  
 → [갈색 블록]에 넣기, [Screen1]의 [방향\_센서1], [방향\_센서1.방위각]  
 → [레이블2]에 넣기



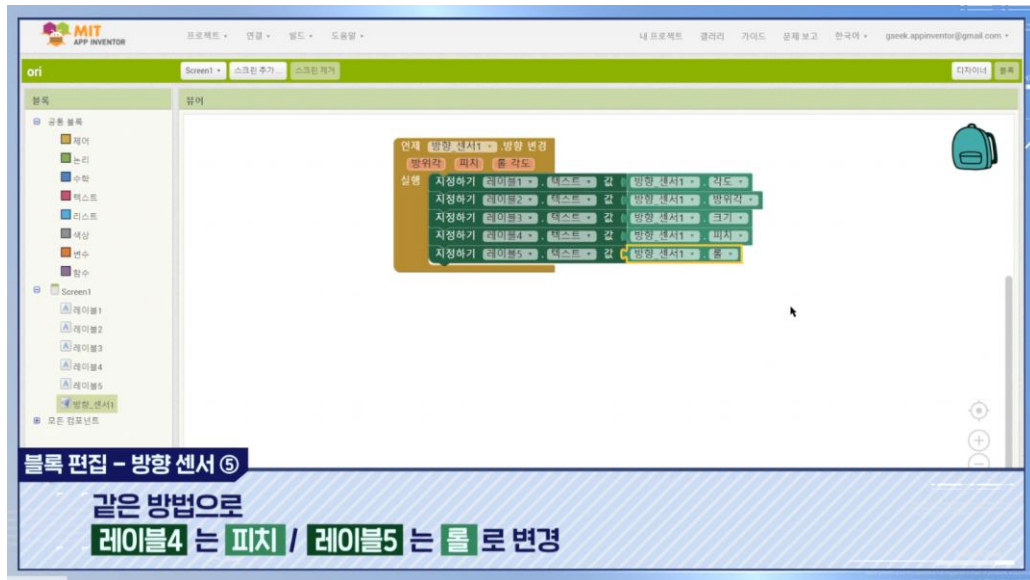
- ④ [지정하기 레이블2.텍스트 값] [방향\_센서1.방위각] 복제 후  
 [갈색 블록]에 넣기, [레이블3] / [크기]로 변경



## 8. 방향 센서를 활용한 공굴리기 앱

⑤ 같은 방법으로 [레이블4]는 [피치] / [레이블5]는 [롤]로 변경

★ 공굴리기 앱의 핵심은 각도와 크기입니다



### 3) 디자이너 편집 - 공 추가

- ① 팔레트 [그리기&애니메이션]의 [캔버스] → 뷰어에 끌어놓기
- ② 속성의 [높이] / [너비] → [부모에 맞추기]
- ③ 팔레트 [그리기&애니메이션]의 [공] → 뷰어에 끌어놓기  
★ 캔버스 안에서만 움직이는 공 컴포넌트입니다
- ④ 속성[페인트 색상]에서 색 변경, 속성[반지름] 변경

## 8. 방향 센서를 활용한 공굴리기 앱

### 4) 블록 편집 - 공 방향 설정

- ① [screen1] [캔버스1]의 [공1] [지정하기 공1.방향 값] → [갈색 블록] 안에 넣기
- ② [방향\_센서1.각도] 복제, [지정하기 공1.방향 값]에 넣기
- ③ [Screen1] [캔버스1]의 [공1], [지정하기 공1.속도 값]  
→ [갈색 블록] 안에 넣기
- ④ 공통 블록의 [수학], [0] → [속도 값]에 놓고 [10]으로 변경
- ⑤ [방향\_센서1.크기] 복제  
★ 스마트폰의 기울어진 정도는 0~1입니다.
- ⑥ 공통 블록의 [수학], [ $\square \times \square$ ] → [속도 값]에 넣기  
★ “두 블록의 곱셈”
- ⑦ [ $\square \times \square$ ]에 [방향\_센서1.크기], [10] 넣기, 속도 [100]으로 변경

### 5) 디자이너 편집 - 목표물 추가

- ① 팔레트 [그리기&애니메이션]의 [이미지 스프라이트] → 뷰어에 끌어놓기
- ② 속성 [사진]의 [없음] → [파일 올리기], [파일 선택] 후 저장된 사진 업로드
- ③ 속성의 [높이] / [너비] 값 변경

### 6) 디자이너 편집 - 알림 추가

- ① 팔레트 [사용자 인터페이스]의 [알림] → 뷰어에 끌어놓기
- ② 속성의 [배경색] / [텍스트 색상] 변경  
★ 특정 사건에만 발생하는 알림 컴포넌트, 사용자 인터페이스에서 유일하게 보이지 않는 컴포넌트입니다

## 8. 방향 센서를 활용한 공굴리기 앱

### 7) 블록 편집 - 목표물 맞추기

- ① Screen1 [캔버스1]의 [이미지\_스프라이트],  
[언제 이미지\_스프라이트1.충돌] 선택
- ② Screen1 [캔버스1]의 [알림1], [호출 알림1.경고 창 나타내기]  
→ [갈색 블록]에 넣기
- ③ 공통 블록의 [텍스트], [“ ”]  
→ [알림]에 넣기, 알림 메시지 공이 닿았어요! 입력  
“이미지에 물체가 충돌하면”+ “경고 창 공이 닿았어요! 발생”