'꼬꼬'의 양계장 탈출

5조 윤서영, 맹수민, 이연우

Contents

- 전체적인 프로젝트 구성
- 시작화면, 성공/실패 화면
- 3가지 게임에 대한 구체적인 디자인 설명

전체적인 구성

- ❖ '꼬꼬'의 양계장 탈출 게임
 - 300초 시간 제한
 - 3단계의 미션 구성
 - 1단계: 퀴즈 형식의 비밀번호 풀기
 - 2단계: 방향센서를 이용해 지붕위로 올라가기
 - 3단계: 버튼 조작으로 다가오는 울타리 피하기

시작화면

STORY





언제 Screen1 ▼ .초기화되었을때 실행 지정하기 이미지1 ▼ . 사진 ▼ 값 page2.jpg ▼ 지정하기 버튼_뒤로가기 ▼ . 활성화 ▼ 값 거짓 ▼ 지정하기 버튼_뒤로가기 ▼ . 보이기여부 ▼ 값 거짓 ▼

 언제 버튼_시작 ▼ .클릭했을때

 실행
 다른 스크린 열기 스크린 이름 【level_1 ▼

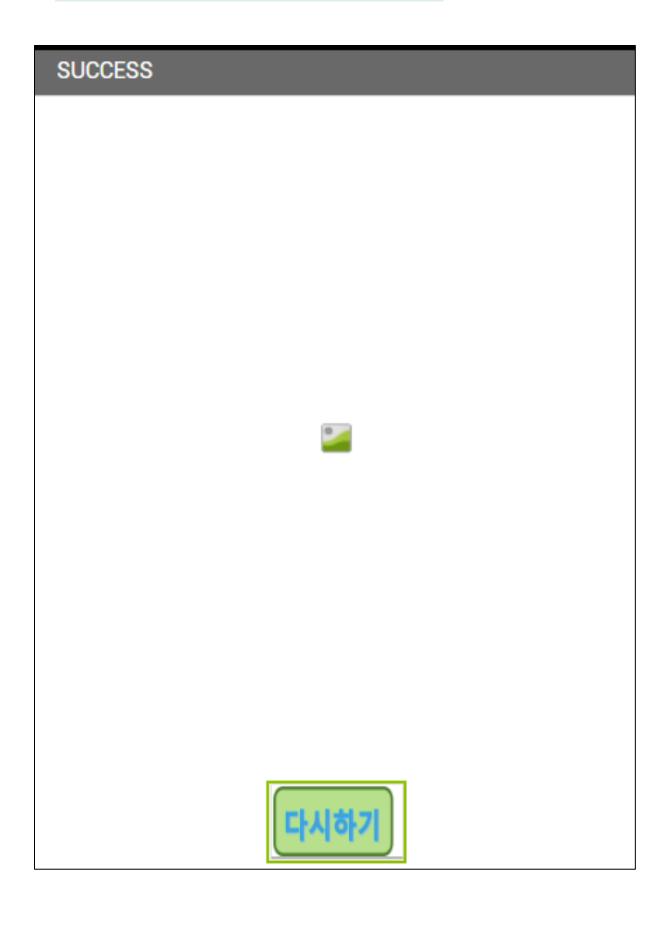
- ✔ 게임에 들어가기 앞서 2컷의 스토리 컷툰 제작
- ✓ 스크린이 초기화 되었을 때 1번째 이미지 보이기
- ✓ 시작 버튼 누르면 1단계 게임 스크린으로 이동

시작화면

```
인제 버튼_뒤로가기 ▼ .클릭했을때
실행 지정하기 이미지1 ▼ . 사진 ▼ 값 ( page2.jpg ▼ 지정하기 버튼_뒤로가기 ▼ . 활성화 ▼ 값 ( 거짓 ▼ 지정하기 버튼_뒤로가기 ▼ . 보이기여부 ▼ 값 ( 거짓 ▼ 지정하기 버튼_다음 ▼ . 활성화 ▼ 값 ( 참 ▼ 지정하기 버튼_다음 ▼ . 보이기여부 ▼ 값 ( 참 ▼
```

- ✓ 다음 버튼을 클릭했을 때
 - 2번째 이미지로 변화
 - 다음 버튼 사라지고 뒤로가기 버튼 생성
- ✓ 뒤로가기 버튼을 클릭했을 때
 - 1번째 이미지로 변화
 - 뒤로가기 버튼 사라지고 다음버튼 생성
 - =〉 버튼의 활성화/보이기 여부 이용

성공화면



언제 버튼_다시하기 · .클릭했을때 실행 다른 스크린 열기 스크린 이름 Screen1 ·

- ✓ 스크린 초기화 되었을 때 성공 이미지 보이기
- ✔ 다시 하기 버튼을 클릭하면 첫 스크린으로 돌아가기

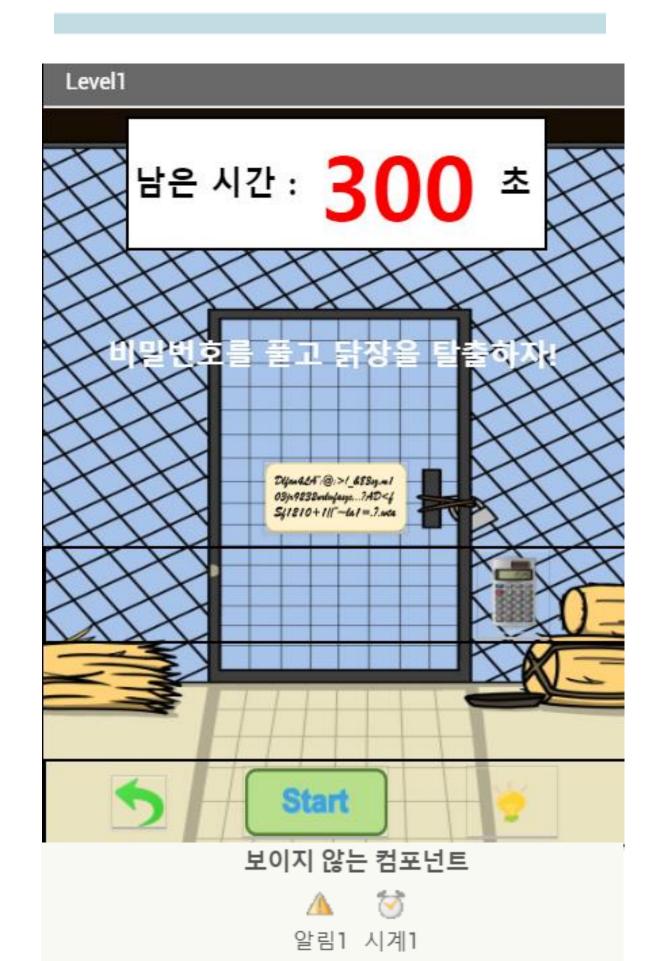
실패화면

FAIL 그래, 너라도 건강해라



- ✓ 실패 이미지 리스트 생성
- ✓ 스크린이 초기화 되었을 때 임의의 이미지 보이기

1단계_시작화면



비밀번호를 풀고 닭장을 탈출하자!

- ✓ 시작 버튼을 클릭하면 타이머 작동, 문제/힌트/키 버튼 활성화
- ✔ 문제 버튼을 클릭하면 문제 스크린으로 이동
- ✓ 비밀번호 키 버튼을 클릭하면 정답 입력 스크린으로 이동
- ✓ 힌트 버튼을 클릭하면 힌트 알림창이 나타남

1단계_시작화면

- ✓ 시작 버튼 클릭 〉 타이머 작동
- = > 레이블 시간 -1씩 감소시켜 0초가 된다면 실패 화면으로 이동

```
언제 버튼_정답입력 ▼ .클릭했을때
   시작 값을 전달하며 다른 스크린 열기 스크린 이름
                            시작 값 (레이블_시간 ▼ . 텍스트 ▼
언제 버튼_문제 ▼ .클릭했을때
   시작 값을 전달하며 다른 스크린 열기 스크린 이름 🔰 🔾 🔻
                                  레이블_시간 ▼
언제 [evel_1 ▼ .다른스크린을닫았을때
 다른스크린이름) (결과)
실행
    지정하기 다른스크린이름 ▼ 값 (🔭 ' Q '
    지정하기 레이블 시간 🔻 . 텍스트 🔻 값 🎁 가져오기 결과 🔻
    지정하기 시계1 ▼ . 타이머활성화여부 ▼ 값 🙀 참 ▼
    지정하기 버튼_문제 ▼ . 활성화 ▼ 값
    지정하기 버튼_힌트 ▼ . 활성화 ▼ 값
    지정하기 [버튼_정답입력 ▼ ]. (활성화 ▼ ) 값 (資本 ▼
```

- ✓ 문제 화면/정답입력 화면을 열고 닫을 때
- =〉시간레이블이 유지되도록 시작 값 전달/받아오기

1단계_문제화면

```
남은 시간 :
            11
           12
           1121
          122111
         000000
         12221131
OOOOOO에 들어갈 숫자는 무엇일까?
         보이지 않는 컴포넌트
            시계1
```

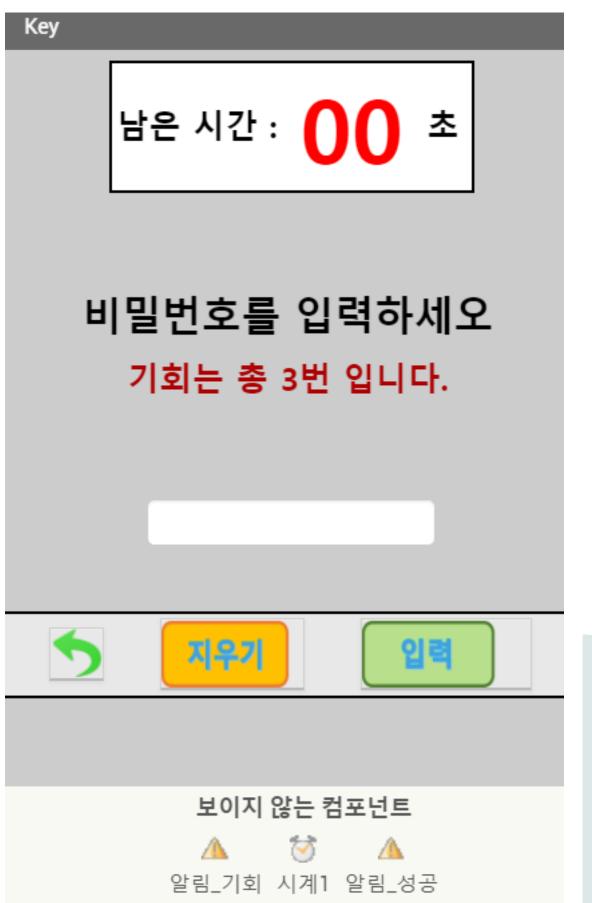
```
언제 Q · 초기화되었을때
실행 지정하기 레이블_시간 · 텍스트 · 값 시작 값 가져오기
지정하기 시계1 · . 타이머활성화여부 · 값 참 ·

언제 시계1 · . 타이머가작동할때
실행 지정하기 레이블_시간 · . 텍스트 · 값 레이블_시간 · . 텍스트 · 1

언제 버튼1 · .클릭했을때
실행 값을 반환하며 스크린 닫기 결과 (레이블_시간 · . 텍스트 ·
```

- ✓ 초기화 되었을 때 시간 레이블 남은 시간 값 가져오기
- ✓ 타이머 동작 유지
- ✔ 뒤로가기 버튼 클릭 했을 때 남은 시간 전달하면 스크린 닫기

1단계_정답입력화면



```
언제 key ▼ .초기화되었을때
   지정하기 레이블_시간 ▼ . 텍스트 ▼ 값 ( 시작 값 가져오기
    지정하기 시계1 ▼ . 타이머활성화여부 ▼ 값 (참 ▼
언제 시계1 ▼ .타이머가작동할때
   지정하기 레이블 시간 🔻 . 텍스트 🔻 값
                              레이블_시간 ▼ . 텍스트 ▼
                                               - (1
언제 버튼_지우기 ▼ .클릭했을때
실행 지정하기 텍스트박스1 🔻 . 텍스트 🔻 값 🚺 0
언제 버튼_뒤로가기 ▼ .클릭했을때
실행 값을 반환하며 스크린 닫기 결과 (레이블_시간 ▼ ). 텍스트 ▼
```

✓ 초기화되었을 때와 시간 유지, 뒤로가기 동작은 문제화면과 동일

✓지우기 버튼 클릭했을 때 텍스트 상자 비우기

1단계_문제화면

전역변수 만들기 (count) 초기값 🚺 3

```
언제 [버튼_입력 🔻 .클릭했을때
   🔯 만약
                가져오기 (전역변수 count ▼ 🕒 ▼
    이라면 실행
                         텍스트박스1 - 텍스트 -
                                               112213
                     호출 알림 성공 🕶 .메시지창보이기
                                      메시지
                                             ' 축하합니다! 다음단계로 넘어가 볼까요?
                                       제목
                                             CLEAR
                                    버튼텍스트
                                             Let's Go
                     시작 값을 전달하며 다른 스크린 열기 스크린 이름
                                                     level2 ▼
                                                    레이블_시간 🔻 . 텍스트 🔻
            아니라면
                     지정하기 전역변수 count 🔻 값
                                            가져오기 전역변수 count *
                     호출 알림 기회 그 .메시지창보이기
                                      메시지
                                            🔯 합치기
                                                     ' [틀렸습니다.]
                                                      ' 앞으로 총
                                                     가져오기 전역변수 count
                                                      ' 기회가 남았습니다.
                                       제목
                                             기회
                                    버튼텍스트
                                             확인
    아니라면
            다른 스크린 열기 스크린 이름
                                fail 🕶
```

- ✓정답 기회 count 전역변수 생성
- ✓ 입력 버튼 클릭 했을 때
- 기회가 남아있고, 정답을 맞췄다면 =〉성공 알림창 보이기, 2단계 화면으로 이동
- 기회가 남아있지만 틀렸다면 => 정답 기회 변수 -1 감소, 틀림 알림창 보이기
- 기회가 남아 있지 않다면 =>실패 화면으로 이동



게임 설명

시작을 누르면 구림의 위치가 임의로 정해진다. 휴대폰을 기울이면 방향센서에 의해서 꼬꼬가

움직인다.

꼬꼬가 하늘에 도달하면 다음 단계로 넘어갈 수 있습니다.

이때 ;!! 구름에 부딪치면 꼬꼬가 튕기게 됩니다!!

```
언제 Screen1 · .초기화되었을때
실행 지정하기 시작 🕶 . 활성화 🕶 값 🎁 참 🔻
    지정하기 닭 🕶 . 🗶 🕶 값
                       캔버스1 ▼ . 너비 ▼ / (2)
    지정하기 닭 ▼ . Y▼ 값 (
                          캔버스1 ▼ . 높이 ▼ × 🕻 0.9
    지정하기 발판3 ▼ . X ▼ 값
                        임의의 정수 시작 (발판3 🔻 년비 🔻 끝
                                                       캔버스1 ▼ . 너비 ▼
                                                                        발판3 🕶 . 너비 🕶
    지정하기 발판4 ▼ . X ▼ 값
                        임의의 정수 시작
                                    발판3 🔻 . 너비 🔻 끝
                                                       캔버스1 ▼ . 너비 ▼
                                                                        발판3 🕶 . 너비 🕶
    지정하기 발판5 ▼ . X ▼ 값
                        임의의 정수 시작 (발판3 ▼ . 너비 ▼ 끝
                                                       캔버스1 ▼ . 너비 ▼
                                                                         발판3 🕶 . 너비 🔻
    지정하기 발판6 ▼ . X ▼ 값
                        임의의 정수 시작 발판3 🕶 . 너비 🕶 끝
                                                                        발판3 ▼ . 너비 ▼
                                                       캔버스1 ▼ . 너비 ▼
    지정하기 발판3 ▼ . Y ▼ 값
                            캔버스1 ▼ . 높이 ▼ × ( 0.2)
    지정하기 발판4 ▼ . Y ▼ 값
                            캔버스1 ▼ . 높이 ▼ × ( 0.4)
    지정하기 발판5 ▼ . Y ▼ 값
                            캔버스1 ▼ . 높이 ▼ ×
    지정하기 발판6 ▼ . ▼ ▼ 값 (
                            [캔버스1 ▼ ]. 높이 ▼ × ( 0.8)
    지정하기 베튼1 🕶 . 보이기여부 🕶 값 🔰 거짓 🔻
```

 스크린이 초기화 되었을 때 닭과 구름의 위치를 지정해준다.

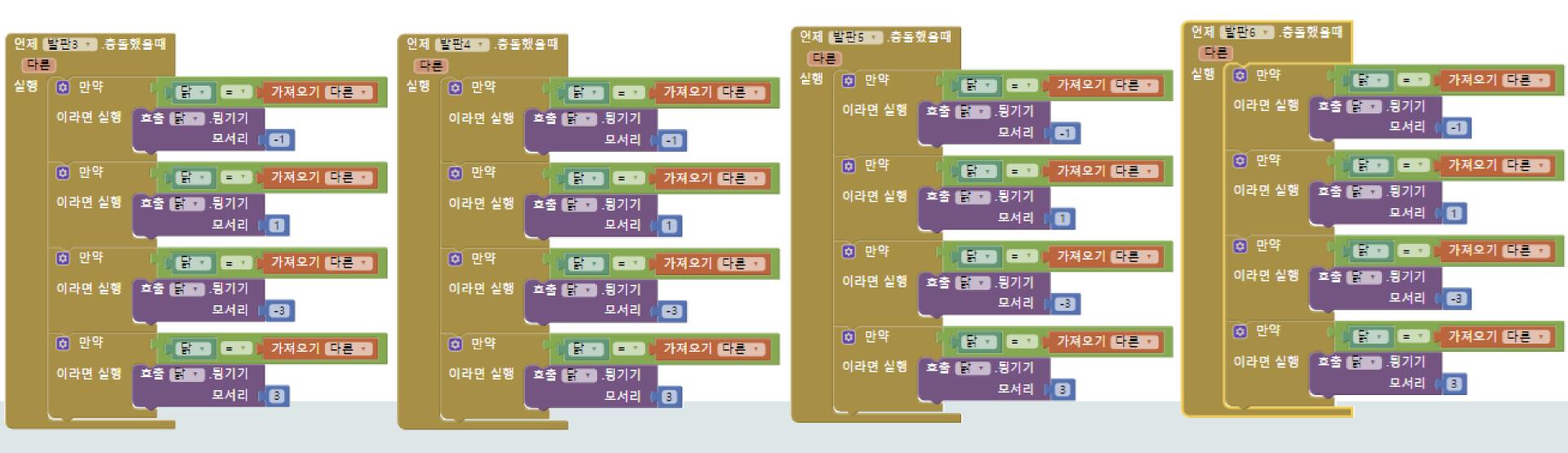
■ 시작 버튼이 작동하도록 활성화한다.

```
언제 시작 .클릭했을때
실행 지정하기 시작 . (활성화 · 값 거짓 ·
    지정하기 닭 ▼ . X ▼ 값 │
                       캔버스1 ▼ . 너비 ▼ / [2]
    지정하기 닭 ▼ . Y▼ 값
                         캔버스1 ▼ . 높이 ▼ × ( 0.9)
    지정하기 닭 🔻 . 시간간격 🔻 값 🔰 2
    지정하기 발판3 ▼ . 🗙 ▼ 값
                        임의의 정수 시작
                                    발판3 ▼
                                           . 더비 🕶 끝
                                                      캔버스1 ▼ . 너비 ▼
                                                                       발판3 🕶 . 너비 🕶
    지정하기 발판4 ▼ . X ▼ 값
                        임의의 정수 시작
                                    발판3 🕶 . 너비 🕶 끝
                                                      캔버스1 🕶 . 너비 🕶
                                                                       발판3 🕶 . 너비 🕶
    지정하기 발판5 🕶 . 🗙 🕶 값
                        임의의 정수 시작
                                    발판3 ▼
                                          . 더비 🔻 끝
                                                      캔버스1 * . 너비 *
                                                                       발판3 🕶 . 너비 🕶
    지정하기 발판6 ▼ . X ▼ 값
                        임의의 정수 시작
                                    발판3 ▼
                                           . 더비 • 끝
                                                      캔버스1 ▼ . 너비 ▼
                                                                       발판3 🕶 . 너비 🕶
    지정하기 발판3 ▼ . Y ▼ 값
                           캔버스1 ▼ . 높이 ▼
                                             0.2
    지정하기 발판4 ▼ . Y ▼ 값
                           캔버스1 ▼ . 높이 ▼
    지정하기 발판5 ▼ . Y ▼ 값
                           캔버스1 ▼ . 높이 ▼
                                             0.6
    지정하기 발판6 ▼ . Y ▼ 값 (
                           캔버스1 ▼ . 높이 ▼ ×
                                             0.8
```

- 시작 버튼을 눌렀을 때 꼬꼬의 위치는 캔버스의 바닥 중앙에 지정해 준다.
- 구름의 y축을 지정해 주지만, x축은 지정해 주지 않아 다양한 패턴이 나오도록 한다.

```
언제 닭 고 .모서리에닿았을때
 모서리
   🤨 만약
실행
               가져오기 모서리 🕶 😑 🕶 🚺 1
    이라면 실행
           지정하기 버튼1 🔻 . 보이기여부 🕶 값 🥻 참 🔻
    🔯 만약
               가져오기 모서리 🔻 🗲 🔻 📜 -1
    이라면 실행
            호출 닭 🔻 .튕기기
                        가져오기 모서리 🔻
                   모서리
언제 시계1 • .타이머가작동할때
    지정하기 닭 🔻 . 속도 🔻 값
                       방향센서1 ▼ . 각도 ▼
    지정하기 닭 🔻 . 방향 🔻 값
언제 [버튼_다시하기 ▼ .클릭했을때
    다른 스크린 열기 스크린 이름
                          Screen1 🔻
```

- 꼬꼬가 캔버스의 모서리에 닿았을 때 튕기도록 설정한다.
- 닭이 하늘에 닿았을 때는 다음 단계로 넘어갈 수 있게 한다.
- 시계 컴포넌트를 넣어서 꼬꼬가 시간 당 움직일 수 있게 한다.
- 뒤로가기를 누르면 게임의 처음으로 넘어간다.



- 닭이 다른 이미지 스프라이트에 닿았을 때 튕기기를 지정해서 꼬꼬가 구름에 닿으면 튕겨서 게임의 재미를 더했다.
- 꼬꼬가 구름의 어느 방향에 닿아도 튕길 수 있도록 했다.



울타리 피하기

꼬꼬가 울타리들을 피해 달리는 장애물 피하기 게임이다.

왼쪽 오른쪽 버튼을 눌러 꼬꼬를 움직여 장애물을 피하기!

남은 거리가 0이 될 때까지 버티면 클리어

```
전역변수 만들기 (남은거리) 초기값 ( 60
전역변수 만들기 (체력) 초기값 📢 🕄
언제 Screen1 ▼ .초기화되었을때
   지정하기 닭 🔻 . 사진 🔻 값 📗 runchick1.png 🔻
    지정하기 체력 . 텍스트  값
                          가져오기 전역변수 체력 🔻
                           가져오기 전역변수 남은거리
    지정하기 거리 🕶 . 텍스트 🕶 값
    지정하기 닭 ▼ . 🗙 ▼ 값
                       캔버스1 ▼ . 너비 ▼ / ( 2
    지정하기 닭 🕶 . 🛛 🕶 값 🛙
                         캔버스1 ▼ . 높이 ▼ × (
    지정하기 울타리 v . X v 값
                         임의의 정수 시작 🕍 🕡 🗎 끝
                                                     . (내 🕶
                                                              울타리 • 너비 •
    지정하기 울타리 🔻 . 🛛 🕶 값
                         © 캔버스1 ▼ . 높이 ▼ ×
                                              0.6
    지정하기 (울타리2 v . X v ) 값
                          임의의 정수 시작 🕡 🕡 끝
                                             캔버스1 🕶
                                                      [년비교
                                                              울타리 🔻 . 너비 🔻
    지정하기 (울타리2 ▼ . Y▼ 값
                             캔버스1 🕶 . 높이 🕶
                                              0.4
    지정하기 울타리2 · . X · 값
                          임의의 정수 시작 🚺 🛈 🗎 끝
                                             캔버스1 • . 너비 •
                                                               울타리 🔻 . 너비 🔻
    지정하기 (울타리 🔻 . 🔞 🕶 값 (
                            캔버스1 ▼ . 높이 ▼ ×
                                              0.1
```

전역변수 '남은 거리'와 '체력'을 설정

꼬꼬의 위치는 화면 하단 중앙에 위치함

장애물이 될 울타리 3개는 각각 y값을 다르게 정하여 서로 다른 높이에 위치하도록 정하지만, x값은 임의로 설정하여 게임을 시작할 때마다 울타리의 위치가 다르게 시작되도록 설정

```
언제 시계 거리 . 타이머가작동할때
   지정하기 울타리 🔻 . 방향 🕶 값
    지정하기 울타리2 🕶 . 방향 🕶 값
                         270
    지정하기 울타리3 🔻 . 방향 🔻 값
                         270
    지정하기 울타리 7 . 속도 7 값
    지정하기 울타리2 🔻 . 속도 🔻 값
    지정하기 울타리3 🔻 . 속도 🔻 값
    지정하기 전역변수 남은거리 🔻 값 🔋
                          가져오기 전역변수 남은거리 🔻
   지정하기 거리 🔻 . 텍스트 🔻 값 👚 가져오기 전역변수 남은거리 🔻
   🧭 만약
               가져오기 전역변수 남은거리 🔻 😑 🔻 🚺 0
    이라면 실행 지정하기 시계 거리 🔻 . [타이머활성화여부 🔻 값 [ 거짓 🔻
```

울타리는 위에서 아래로 떨어짐

시계 컴포넌트를 사용하여 시계_거리

컴포넌트를 만들었음

타이머간격을 1000으로 설정하고, 매 초마다

남은 거리가 1씩 줄어들게 함

거리가 0이 되면 게임을 클리어하여 화면을 넘어감

```
언제 울타리 7 .모서리에닿았을때
 모서리
실행 지정하기 울타리 🕶 . 🗶 🕶 값
                        임의의 정수 시작
                                    (0) 끝
                                            캔버스1 🔻 . 너비 🔻
                                                             울타리 🔻 . 너비 🔻
    지정하기 울타리 🔻 . Y 🕶 값
                           울타리 ▼ . 높이 ▼ ×
                                            1.1
언제 울타리2 . 모서리에닿았을때
 모서리
실행 지정하기 울타리2 🕶 . 🗶 🕶 값
                          임의의 정수 시작 (*
                                     (0) 끝
                                                              울타리2 🔻 . 너비 🔻
                                             캔버스1 🔻 . 너비 🔻
    지정하기 울타리2 🕶 . Y 🕶 값 🛭
                             울타리2 ▼ . 높이 ▼ × ( 1.1
 언제 울타리3 🔻 .모서리에닿았을때
  모서리
     지정하기 울타리3 🕶 . 🗙 🕶 값
                          임의의 정수 시작 (0 끝
                                              캔버스1 🔻 . 너비 🔻
                                                               울타리3 🔻 . 너비 🔻
     지정하기 (울타리3 🔻 . Y 🕶 값
                             울타리3 🔻 . 높이 🔻
```

울타리가 바닥까지 내려오면 다시 위쪽으로 보내도록 y값을 설정

X값은 임의의 숫자가 나오게 하여 울타리가 다시 나타나는 위치는 매번 다르게 됨

```
이부분
  시계_닭 🔻 .타이머가작동할때
   🔯 만약
                닭 7 . 사진 7 = 7
                               runchick1.png 🔻
    이라면 실행 지정하기 닭 🔻 . 사진 🕶 값
                               runchick2.png
    아니고 만약
                닭 🕶 . 사진 🕶 📒 🔻
                                runchick2.png
                               runchick1.png
   아니고 만약
                닭 🔻 . 사진 🔻 😑 🔻
                                crushchick.png
   이라면 실행 지정하기 닭 🔻 . 사진 🔻 값 🙀 runchick1.png
언제 인쪽 🔻 .클릭했을때
   지정하기 닭 🕶 . 🗙 🕶 값
                       닭 v . X v
                                    캔버스1 🔻 너비 🔻
언제 오른쪽 ▼ .클릭했을때
                         닭 7 . X 7 +
                                       캔버스1 • 너비 •
                                                       10
                        〈-사용된 꼬꼬의 달리기 이미지
```

시계 컴포넌트로 꼬꼬가 움직이는 것처럼 보이게 함

타이머 간격을 500으로 설정하고 사진이 계속 바뀌게 하여 꼬꼬가 달리는 모습을 만듬

꼬꼬의 위치 이동은 왼쪽, 오른쪽 버튼을 터치하면 일정 거리를 버튼에 맞춰 좌우로 이동하게 함

언제 울타리 🔻 .충돌했을때

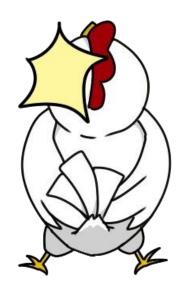
```
다른
실행 🔯 만약
              호출 울타리 🔻 .충돌여부가져오기
    이라면 실행 지정하기 전역변수 체력 🔻 값 🚺 가져오기 전역변수 체력 🔻
                                 가져오기 전역변수 체력
            지정하기 닭 🔻 . 사진 🔻 값 🌘 crushchick.png 🔻
                가져오기 전역변수 체력 🕶 😑 🕶 🗓 0
    이라면 실행 | 다른 스크린 열기 스크린 이름 | gameover 🔻
언제 (울타리3 ▼ .충돌했을때
 다른
실행 🔯 만약
             호출 울타리3 🔻 .충돌여부가져오기
    이라면 실행 지정하기 전역변수 체력 🔻 값 기저오기 전역변수 체력 🔻
                                가져오기 전역변수 체력
            지정하기 체력 🕶 . 텍스트 🕶 값 🌘
                가져오기 전역변수 체력 🕶 😑 🕶 🗓 0
            다른 스크린 열기 스크린 이름 🕽 gameover 🔻
언제 울타리2 🔻 .충돌했을때
다른
실행 🔯 만약
              호출 울타리2 🔻 .충돌여부가져오기
   이라면 실행 지정하기 전역변수 체력 🔻 값 🚺 가져오기 전역변수 체력 🔻
   🔯 만약
                가져오기 전역변수 체력 🕶 😑 🕶 🕻 0
   이라면 실행 다른 스크린 열기 스크린 이름 gameover
```

꼬꼬가 울타리에 충돌하면 전역변수의 체력이 1만큼 줄어들며, 꼬꼬의 이미지 스프라이트가 잠시 충돌한 이미지로 바뀜 충돌한 이미지로 바뀌어 있는 시간은 시계_닭 컴포넌트를

```
언제 시계 닭 ▼ .타이머가작동할때
실행 만약 당 ㆍ 사진 ▼ = ▼ runchick1.png ▼
이라면 실행 지정하기 닭 ▼ . 사진 ▼ 값 runchick2.png ▼
이라면 실행 지정하기 닭 ▼ . 사진 ▼ 값 runchick1.png ▼
이나고 만약 당 ▼ . 사진 ▼ 값 runchick1.png ▼
이라면 실행 지정하기 닭 ▼ . 사진 ▼ 값 runchick1.png ▼
```

이용하여 0.5초로 설정함

체력이 0이 될 시 게임오버



이 부분