

개발 프로그램 설명서

※ 프로그램 제목

달려라 우리반!

※ 프로그램을 만들게 된 이유나 계기를 써보세요.

제주도의 비만도는 **전국 1위**입니다. 그래서 저희 학교에서 매주 목요일마다 운동장 3바퀴씩 달리기를 하고 있습니다. 하지만 운동장을 제대로 달리지 않고 재미없어하는 친구들이 많이 있습니다. 저희는 더 재미있게 달리기를 할 수 있는 방법은 없을까 생각하던 중 친구들이 자주하는 보드게임 형식으로 하면 어떨까 라는 생각을 하게 되었습니다. 그래서 저희는 비만 문제를 해결하기 위해서 친구들이 달리기를 잘 뛰게 하기 위해서 보드게임 형식으로 앱을 만들게 되었습니다.

※ 어떤 프로그램인지 그림이나 글로 설명하세요.

처음 스크린에 설정, 시작, 공유, 랭킹버튼 4개가 있습니다.

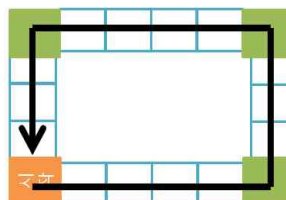


설정 버튼을 클릭하면 반 이름, 인원수, 상품 등의 정보 입력란과 게임 주의사항이 나오게 됩니다. 설정에서 입력한 정보를 바탕으로 게임이 진행됩니다. 몇 조를 입력하느냐에



따라 게임에서 진행되는 말의 개수가 달라지게 됩니다.

게임은 보드게임 형식으로 진행되고 조의 모든 학생이 각자 두 바퀴씩 운동장을 뛰었다는 체크를 했을 때 말이 한 칸씩 움직입니다.



보드게임 한 바퀴를 돌게 되면 해당 조는 설정에서 학급이 함께 정한 보상을 받게 됩니다. 각 조의 점수는 랭킹에 올라가고 **★ 랭킹**, 앱이 어떻게 진행되는지 도움말을 넣어 궁

금한 것은 알게 했습니다. 함께하면 행복이 배가 되듯이 반 친구들 모두가 저희의 **달려라 우리반!** 프로그램을 통해 보다 즐겁고, 열심히 운동을 할 수 있을 것입니다. 또한 꾸준한 운동으로 비만을 예방하고, 건강한 생활을 할 수 있습니다.

※ 프로그램의 동작원리나 알고리즘을 작성해보세요.



6학년 1반 친구들 30명이 다같이 설정을 했다. 조는 총 6조로 하였다.

그 다음 게임화면으로 가서 누르세요 버튼을 누르고, 각자 운동장 2바퀴씩 즉 한 조당 10바퀴를 뛰었다. 게임화면에 다시 들어가 10바퀴를 뛰었다고 입력하였다. 그 다음 날에는 1조, 2조, 3조, 5조 만 각자 2바퀴 씩 뛰고, 4조와 6조는 다 뛰지 못했다. 2주일이 지난 후 1, 3, 5조는 설정에서 정한 보상을 받게 되었다. 그럼 이제 보상을 받은 친구들의 후기를 듣도록 하겠습니다.

- **김00**: 안녕하세요. 저는 000초등학교를 다니는 김00입니다. 저는 달려라 우리반! 이라는 앱을 사용하면서 운동을 평소보다 더 많이 할 수 있게 되었습니다. 하루에 2바퀴씩 뛰지 못하면 다른 조한테 뒤처지기 때문이죠. 저는 보상을 얻기 위해서 2주일동안 꾸준히 운동을 할 수 있었습니다.
- **최00**: 안녕하세요. 저는 비록 보상을 따지 못했지만 교훈을 얻을 수 있었습니다. 다음부터는 꼭 운동을 열심히 할 거라고 결심을 하게 되었습니다. 저에게는 달려라 우리반이란 앱은 운동의 시작이 되었습니다.

※ 프로그램 제작시 역할

| 팀원 | 역할 |
|-----|------------|
| ○○○ | 보고서 작성과 발표 |
| ○○○ | 앱 코딩 및 수정 |
| ○○○ | 디자인, 앱 코딩 |
| ○○○ | 앱 코딩 및 수정 |

개발 프로그램 설명서

※ 프로그램 제목

Bicycle For Safe

※ 프로그램을 만들게 된 이유나 계기를 써보세요.

많은 학생들과 직장인들은 등교 혹은 출근을 할 때 자전거를 타고 다닌다. 대부분의 사람들은 안전장비를 모두 착용하지 않거나 횡단보도에서 자전거를 끌고 가지 않는다. 또한, 차가 다니지 않는다고 도로에서 자전거를 타는 사람들도 있다. 이는 자신뿐만 아닌 다른 사람들도 위협하는 행위이다. 물론 이를 대비하는 기본적인 방법은 자전거를 타는 사람의 실천이지만 이것만으로 안심하기는 이르다. 우리 팀은 이 프로그램을 만들며 사람들이 인식하지 못하는 부분도 소프트웨어와 하드웨어로 인식하여 자전거 사고를 방지하기 위해서 제작하게 되었다.

※ 어떤 프로그램인지 그림이나 글로 설명하세요.

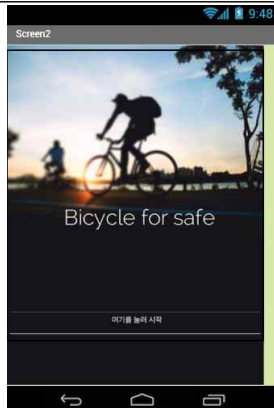


- Bicycle_for_safe 아이콘

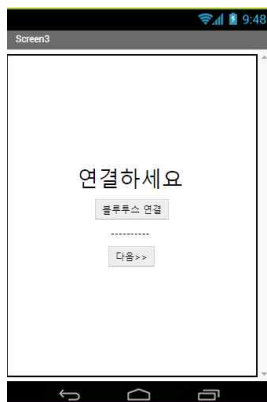
(이 아이콘이 자전거와 관련된 앱이라는 것을 시각적으로 표현해준다.)



- 로고 (우리 팀을 상징하는 로고가 화면에 나타나는 장면이다.)

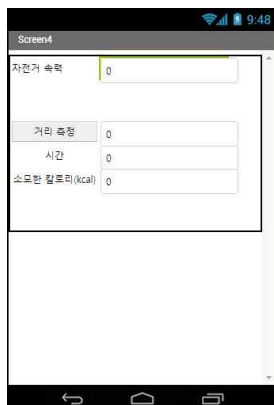


- 시작 화면 (제목과 시작 버튼이 화면에 나타나는 장면이다.)



- 블루투스 연결 화면

(자전거에 부착된 비트브릭을 블루투스로 연결해 정보를 얻을 수 있도록 블루투스 기능을 실행시키는 화면이다.)



- 칼로리, 자전거 속도, 거리, 시간 측정 화면

(자전거를 타고 소모한 칼로리, 자전거 속도, 거리, 시간을 계산해 화면에 구체적 수치로 나타낸다.)

※ 프로그램의 동작원리나 알고리즘을 작성해보세요.

1.비트브릭

가.ir센서

- 1) 자전거 앞쪽 포크(앞바퀴를 고정 시켜주는 막대)한쪽에 ir센서를 부착한다.
- 2) 자전거 바퀴살(바퀴에 있는 쇠창살)에 스티커를 부착한다.
- 3) 자전거 바퀴의 원주를 구한다.($S = 2\pi r$)
- 4) ir센서를 이용해 스티커가 한바퀴 돌고 다시 원위치로 돌아올때까지 소요되는 시간을 구한다.
- 5) 자전거의 속력을 구한다.(속력=거리/시간)

나. LED센서

- 1) 앞에서 ir센서를 이용해 구한 속력에 따라 LED색상을 바꾼다.
- 2) 속력 ≤ 15 LED색상: 초록색
- 3) $15 < \text{속력} \leq 20$ LED색상:노란색
- 4) $20 < \text{속력}$ LED색상:빨간색

다. 버튼

- 1) 버튼을 누르면 자전거의 속도를 줄인다.(급정지)

라. 모터

- 1) 버튼을 누를 때: 모터를 이용해 자전거의 브레이크 역할을 한다.
- 2) 속도가 20을 넘겼을 때 : 모터를 이용해 속도를 20 이하로 내린다.
- 3) 원리: 자전거의 브레이크와 동일하게 모터를 이용해 자전거 바퀴의 림을 약하게 눌러 준다.

2.앱 인벤터

가. 비트브릭과 연결

- 1) 블루투스를 이용해 비트브릭과 연결한다.
- 2) 실패 원인: 블루투스 모듈의 고장으로 인해 연결실패

나. 거리측정

- 1) GPS센서를 이용하여 시작할때의 위도와 경도를 구한다.
- 2) GPS센서를 이용하여 끝났을때의 위도와 경도를 구한다.
- 3) 위도와 경도를 이용해 거리를 구하는 공식(너무길어서 생략)을 이용하여 이동거리를 구한다.

다. 칼로리 측정

- 1) 칼로리를 측정하는 공식을 이용하여 이동한거리에 해당하는 칼로리를 구한다.
- 2) 몸무게(kg)*평균속도별 칼로리 계수*분(시간)=소모된 칼로리(Kcal)
- 3) 평균 속도별 칼로리

| 평속(km/h) | 칼로리소비계수(Kcal/kg/분) |
|----------|--------------------|
| 13 | 0,0650 |
| 16 | 0,0783 |
| 19 | 0,0939 |
| 22 | 0,113 |
| 24 | 0,124 |
| 26 | 0,136 |
| 27 | 0,149 |
| 29 | 0,163 |
| 31 | 0,179 |
| 32 | 0,196 |
| 34 | 0,215 |
| 37 | 0,259 |
| 40 | 0,311 |

※ 프로그램 제작 시 역할

| 팀원 | 역할 |
|-----|---------------------|
| ○○○ | 비트브릭 프로그래밍&프로그램 설명서 |
| ○○○ | 앱 인벤터 프로그래밍&아이콘 제작 |
| ○○○ | 그래픽 디자인 담당&프로그램 설명서 |
| ○○○ | 자료 조사&ppt제작 |

개발 프로그램 설명서

※ 프로그램 제목

bi itself (바이 잇셀프. **b**icycle takes care of **i**tself. 자전거가 알아서 한다.)

※ 프로그램을 만들게 된 이유나 계기를 써보세요.

자전거 사고는 지난 5년간 연 평균 1만 4475건 발생했으며, 그중 275명이 사망하고 1만 2358명은 부상을 입었습니다. 연평균이 1.1% 씩 올라가는 자동차 사고에 비해 자전거는 연평균 9.4% 씩 증가하며, 3차선 이상의 큰 도로에서 41.2% 의 사고가 발생합니다.

사람들은 대개 자전거 사고에 대해 가볍게 생각하는 경향이 있습니다. ‘모터 달린 것도 아닌데 뭐’ ‘사고가 나도 사람이랑 부딪치는 정도겠지’ 하지만 실상은 전혀 그렇지 않습니다. 자전거 사고 중 91.8%는 차와 차, 즉 자전거와 자동차가 충돌해 나는 사고입니다. 이런 사고는 많은 인명피해를 내며, 2차 3차사고까지도 이어질 수 있습니다.

이렇게 위험한 자전거 사고는 왜 일어나는 것일까요? 여러 이유가 있지만, 그중 가장 큰 비중을 차지하는 것은 복잡한 차도에서 빠르게 튀어나오는 차량들을 보지 못하기 때문입니다.

3,4차선처럼 복잡한 도로에 수많은 차들은 각각 다른 여러 방향으로 주행하는데, 이 차들을 자전거 운전자가 모두 보고 예측할 수 있는 것이 한계가 있습니다. 이런 이유들이 겹쳐 복잡한 도로에서 자전거 사고가 자주 일어났던 것입니다.

저희는 이런 점이 자전거를 안전하게 탈 수 없게 하는 가장 큰 요인이라고 생각했고, 어떻게 하면 더 안전하게 자전거를 탈수 있을지 생각했습니다. 자동차와 자전거가 차도에서 서로 부딪치는 것을 어떻게 해야 막을 수 있을까?

저희는 팀원과의 수많은 토의를 통해 서로의 의견을 종합했고 아이디어를 완성할 수 있었습니다. 자전거 운전자가 차들을 잘 볼 수 없다면, 자전거가 (프로그램이) 알아서 해줄 수 있지 않을까? 하는 생각이었습니다. 그렇게 만들어 진 것이 바로 ‘바이 잇 셀프’ 입니다. 먼 곳에서 오는 차들을 프로그램이 인식 후, 차가 근방에 다가왔을 때 운전자에게 조심하라고 알려주는 기능을 가지고 있습니다. 또한 밤에도 안전하게 탈 수 있는 자전거를 위해 평범한 LED 등이 아닌 자동차의 방향 지시등의 역할을 할 수 있도록 LED도 새롭게 바꿨습니다.

자전거 운전자가 볼 수 없는 곳에서 오는 위험도 막아주는 프로그램. 이것이 바로 저희의 ‘바이 잇 셀프’가 나온 계기입니다.

※ 어떤 프로그램인지 그림이나 글로 설명하세요.

‘바이 잇 셀프’는 이름 그대로 운전자가 신경 쓰지 않아도 자전거가 알아서 해주는 프로그램으로, 피지컬 컴퓨팅 중 초음파센서와 버튼, 레인보우 LED, LCD보드, 조도센서를 사용했습니다. 먼저 자전거 측면에 부착된 초음파센서를 통해 물체를 감지 후, 몇 초 뒤 다시 거리를 재는 방식을 사용해 물체의 속도를 측정하여 이것이 운전자에게 위협이 될 만한 것 (자동차, 오토바이 등) 인지 아닌지 감지합니다. 만약 그것이 느린 물체(벽, 사람 등)라면 반응하지 않고, 빠른 물체라면 자전거의 일정거리 내로 들어올 시 LCD 보드가 작동됩니다. LCD 보드에는 왼쪽 혹은 오른쪽에서 몇m 안에 빠르게 오는 물체가 있다는 정보가 나타납니다.

이 시스템을 통해 평범한 초음파 센서 하나로 자전거에게 안전한 물체는 표시하지 않고, 위협을 주는 빠른 물체가 어느 쪽에서 몇미터 근처로 오고 있는지 자전거 운전자에게 알려줄 수 있습니다.

또한 밤에 일어나는 자전거 사고를 막기 위해 밤이 되었을 경우 이를 조도센서가 감지, 자전거 앞에 설치되어 있는 LED를 자동적으로 켜줍니다. 뒷부분의 LED는 낮에도 켜져 있습니다.

평범한 LED 등보다 더 안전하게 탈 수 있도록 뒤에 설치된 등온 자동차의 방향지시등 역할을 할 수 있게 프로그래밍 하였습니다. 자전거 양쪽 손잡이에 있는 버튼중 하나를 누르면 평범히 빛나던 등이 버튼을 누른 쪽을 가르키는 화살표를 표시하는 것입니다(ex, 오른쪽 버튼을 눌렀을 경우->뒤에 등이 오른쪽 화살표로 변함.)

초음파 센서를 사용해 안전 사각지대를 미연에 방지하고, 방향지시등 역할을 하는 LED로 안전을 더하는 프로그램. ‘바이 잇셀프’입니다.

※ 프로그램의 동작원리나 알고리즘을 작성해보세요.

프로그램이 시작되면, 우선 자전거 전면 양쪽에 비스듬히 달린 두 개의 초음파 센서가 전방의 장애물을 감지한다. 만약 전방에 장애물이 있다면, 그 장애물의 초당 이동거리를 측정하여 달리는 차인지 여부를 확인한다. 만약 전방의 장애물이 달리는 차라면(초당 이동거리가 길다면), 차가 자전거로부터 80cm 이내에 있을 때 자전거 운전자에게 LCD 보드로 차와의 거리를 알려준다. 전방의 장애물이 달리는 차가 아니라면 LCD 보드에 아무런 값도 뜨지 않는다.

또 하나의 기능으로, 조도 센서의 입력값을 판단하여 현재가 낮인지 밤인지 알아낸다. 만약 현재가 밤이라면, 자전거의 전조등(레인보우 LED)을 켜다. 그리고 자전거 손잡이에 달린 버튼이 눌리면, 눌린 버튼이 왼쪽 버튼일 경우 후미등에 왼쪽 화살표 표시를 하고, 오른쪽 버튼은 오른쪽 화살표 표시를 한다.

※ 프로그램 제작시 역할

| 팀원 | 역할 |
|-----|-------------------------|
| ○○○ | 프로그래밍, 아이디어 구상, 모형 제작 |
| ○○○ | 프로그래밍, 아이디어 구상, 모형 제작 |
| ○○○ | 아이디어 구체화, 설명서 작성, 모형 제작 |
| | |