

오픈소스기초설계

Halli Galli Game

1팀
명준모
곽창민
김재은

CONTENTS

01 배경 및 동기 02 작품 소개

03 작품 구현 04 보완 & 효과

CONTENTS

01 배경 및 동기

대중적인 보드게임



- 사람들이 즐겨 찾는 대중적인 보드게임
- 라즈베리 파이와 LED matrix를 이용해 구현할 수 있는 저사양의 게임

카드를 뒤집는 방향



- 카드는 다른 플레이어들이 먼저 볼 수 있게 뒤집어야 함
 - 모든 플레이어가 동시에 카드를 볼 수 없음
- ➔ 모든 플레이어가 동시에 볼 수 있도록 컴퓨터가 일정 주기로 카드를 뒤집어 보여줌

종의 개수

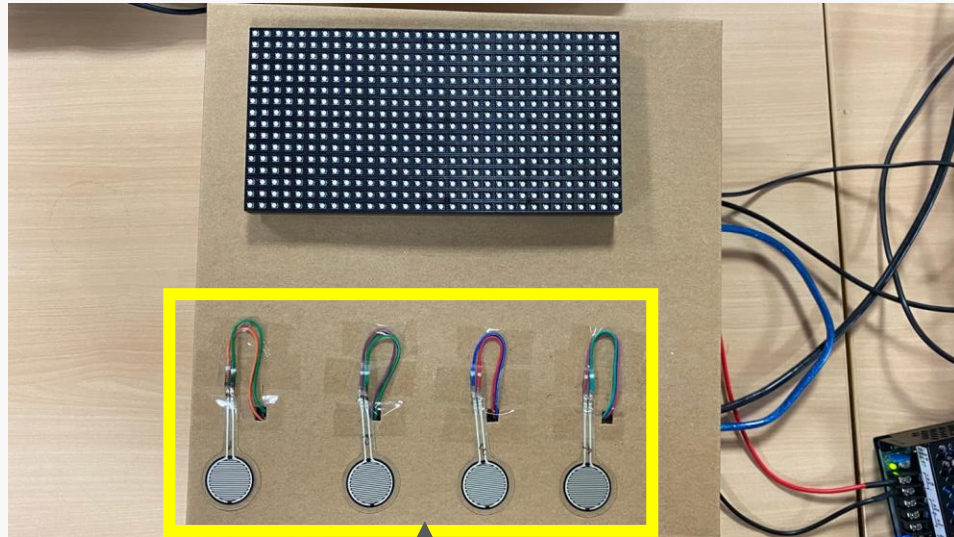
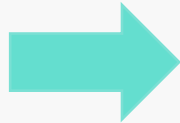


- 다수의 플레이어가 하나의 종을 침
 - 승패를 가리기 어려운 상황이 발생할 수 있음
(ex. 두명의 손이 겹쳐져 종을 친 상황)
- ➔ 플레이어에게 각각의 종을 부여하여 가장 빨리 누른 플레이어를 컴퓨터가 판단함

02 작품 소개

❖ 기존 할리갈리 게임보다 더 공정하게 구현

- 사람이 아닌 컴퓨터가 게임의 승패 판정
- 압력센서를 이용하여 플레이어마다 각자의 종을 부여함
- 컴퓨터가 1초마다 카드를 뒤집어서
모든 플레이어가 동시에 카드 진행 상황을 볼 수 있음

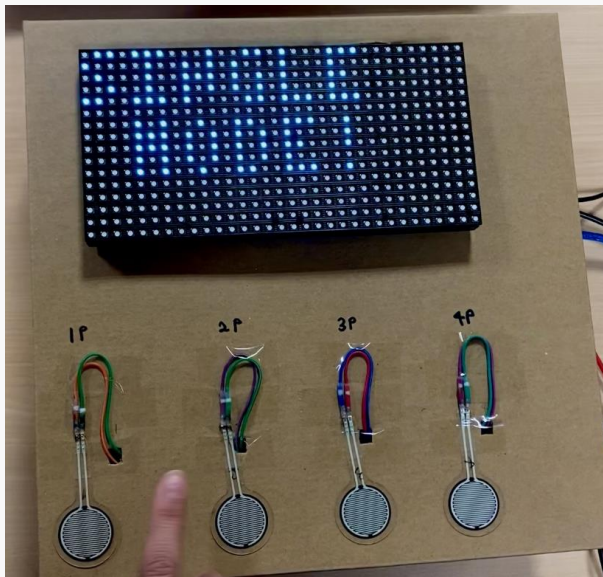


플레이어마다 가지는 종(압력센서)

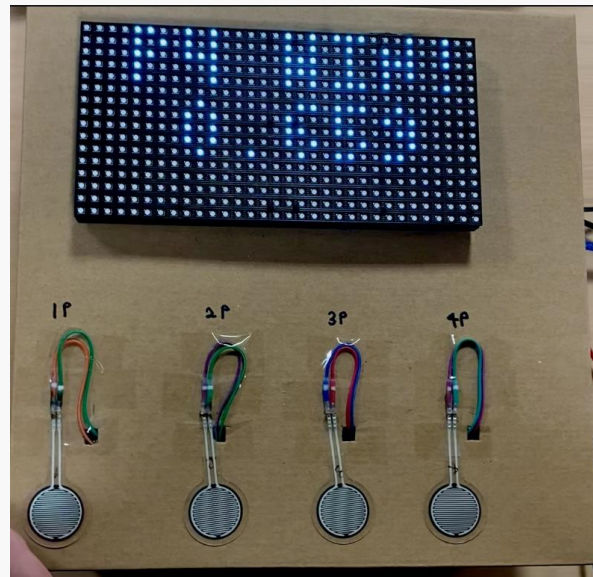
02 작품 소개

❖ 추가 기능

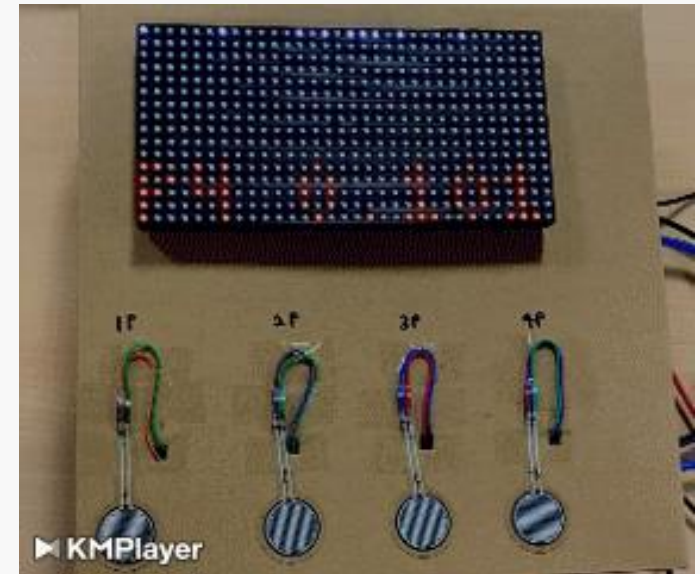
- Single mode : 한 명의 플레이어와 컴퓨터의 대전
- 종을 칠 때마다 반응 시간 체크 → 반응시간 비교로 정확한 승패 판단 가능
- 반응 시간 TOP 10 랭킹 → 플레이어의 경쟁심 & 흥미 유발



Single mode



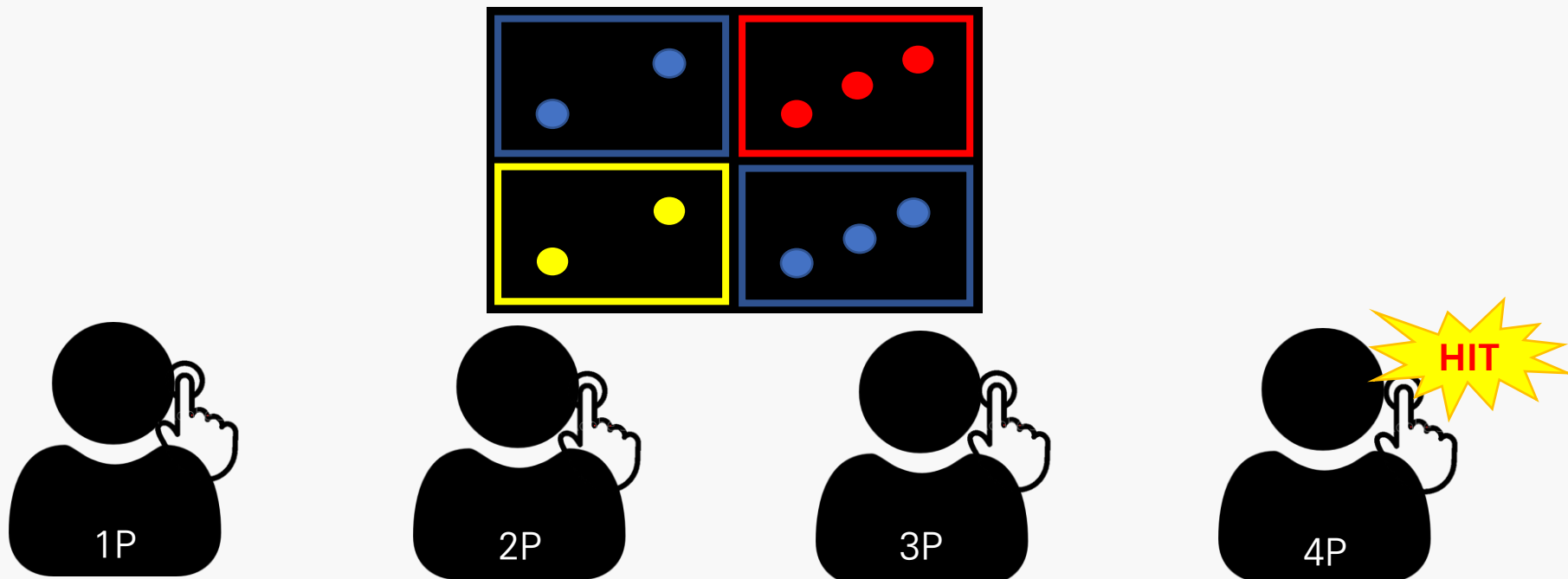
반응시간 체크



TOP 10 랭킹

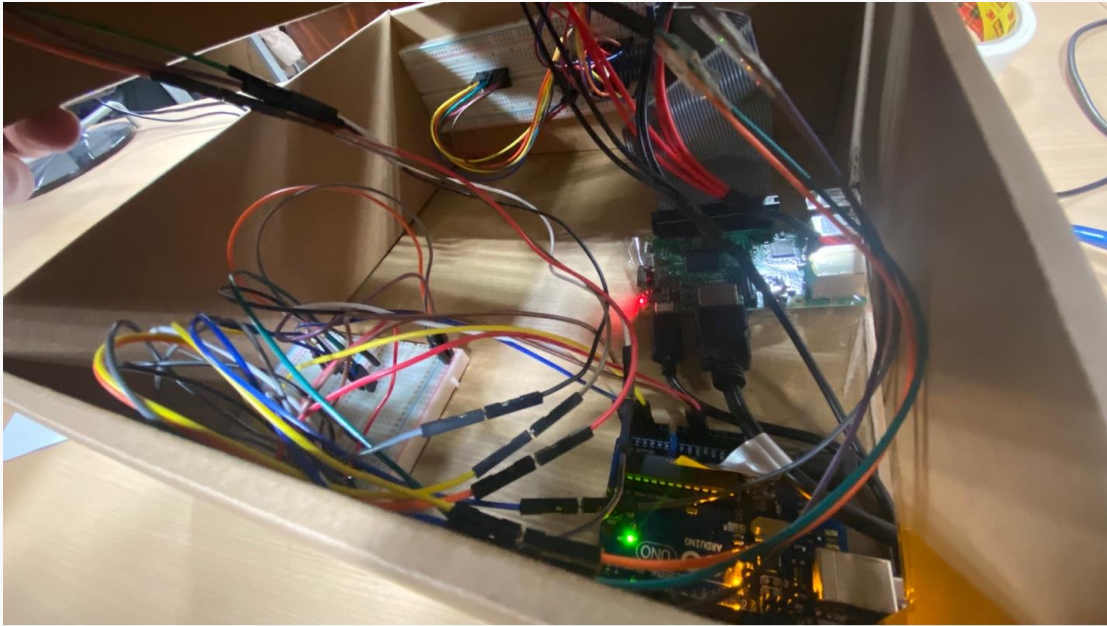
03 작품 소개

- ◆ 카드의 종류는 네가지 색상으로 나누었으며 각 카드는 1~5개의 동그라미를 가짐
- ◆ 놓여진 카드들 중 같은 색상의 동그라미가 5개일 경우 종을 쳐야 함
- ◆ 올바르게 종 칠 경우 놓여진 카드들 모두 획득 / 잘못된 경우 카드를 나머지 사람에게 나눠줌
- ◆ 카드는 56장 이며 4인 게임인 경우 14장씩, 싱글모드인 경우 28장씩 섞어서 카드 제공
- ◆ 카드는 1초마다 컴퓨터가 보여줌!
- ◆ 종은 4개의 압력센서를 통해 각 플레이어에게 하나씩 제공함

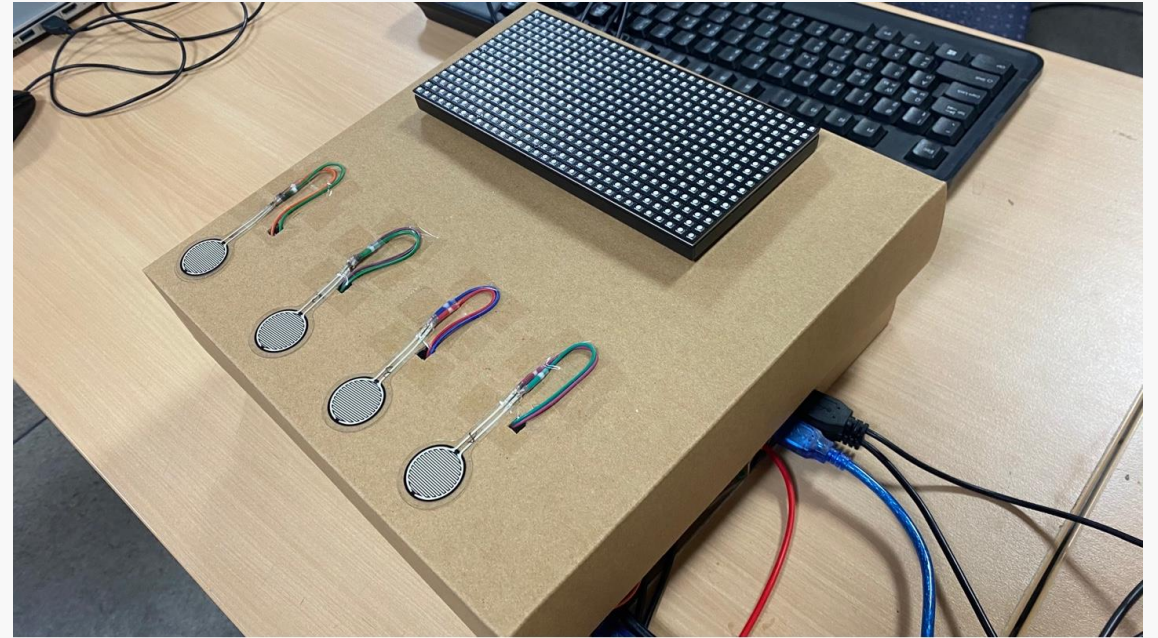


02 작품 소개

❖ 구현 작품 구조



내부 구조



외부 구조

03 작품 구현 - 로직 수정

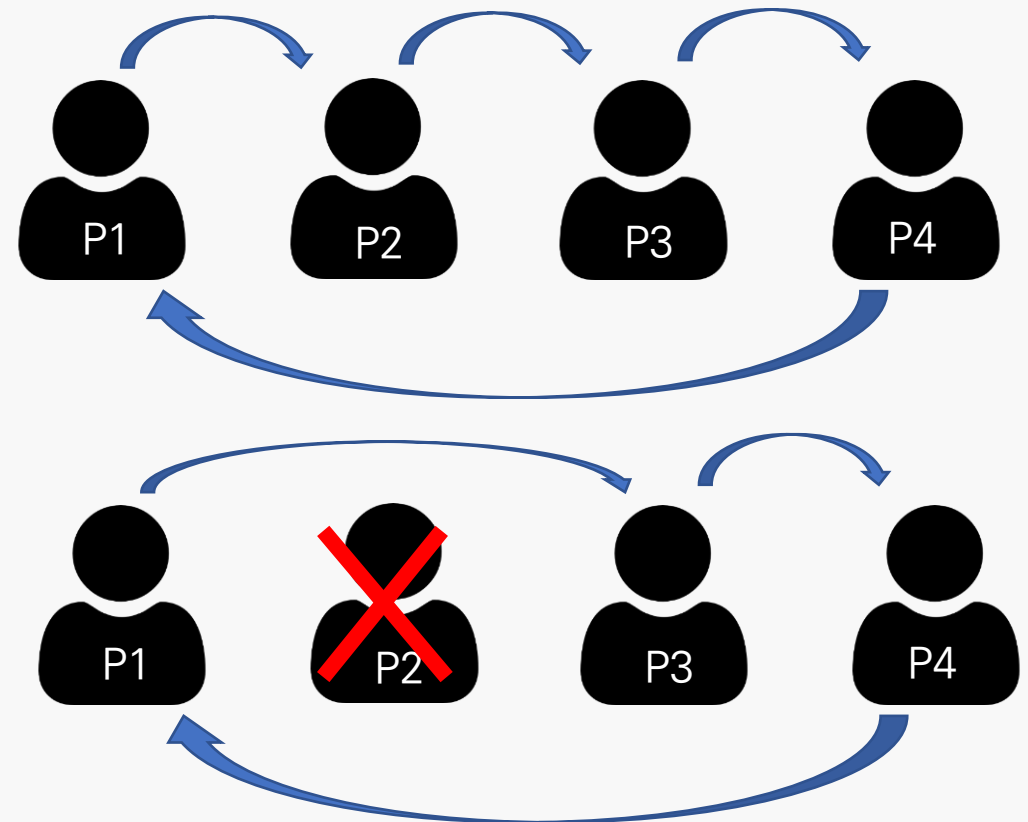
- ◆ 오픈소스 할리갈리 게임은 키보드 입력으로 사용자가 직접 카드를 뒤집음
→ 본 게임은 컴퓨터가 자동으로 카드를 뒤집어야 하기 때문에 게임 로직 수정 필요!

4P mode

순서: P1-P2-P3-P4

(살아있는 플레이어들에 한해 반복)

1. 현재 플레이어의 카드 뒤집기
2. 뒤집힌 카드들 중 5개가 된 카드가 있는가 판단
3. 1초 안에 플레이어들의 센서 값 읽기
(있다면 종을 친 상황)
 - 4-1. 모두 종 안침: 다음 사람 순서로
 - 4-2. 종 침: 제대로 쳤다면 반응 시간 계산!
 - 다음 플레이어 시작
 - 잘못쳤다면 패널티 부여!
 - 종 친 사람 부터 시작



03 작품 구현 - 로직

◆ Single mode: P1(사용자)과 P4(컴퓨터) 간 반응 시간 대결

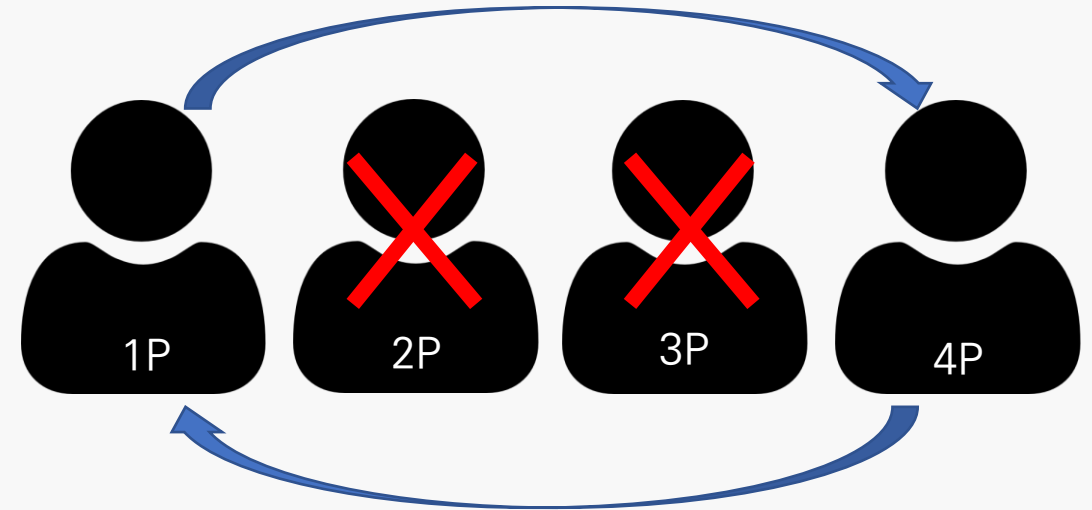
- 컴퓨터는 HARD, NORMAL, EASY 난이도를 가지며 난이도에 따라 종을 치는 반응 시간이 다름

1P mode

순서: P1-P4

(살아있는 플레이어들에 한해 반복)

- 현재 플레이어의 카드 뒤집기
- 뒤집힌 카드들 중 5개가 된 카드가 있는가 판단
- 1초 안에 플레이어들의 센서값 읽기
(있다면 종을 친 상황)
 - 모두 종 안침: 다음 사람 순서로
 - 종 침: 제대로 쳤다면 반응 시간 계산!
 - 다음 플레이어 시작
 - 잘못쳤다면 패널티 부여!
 - 종 친 사람 부터 시작

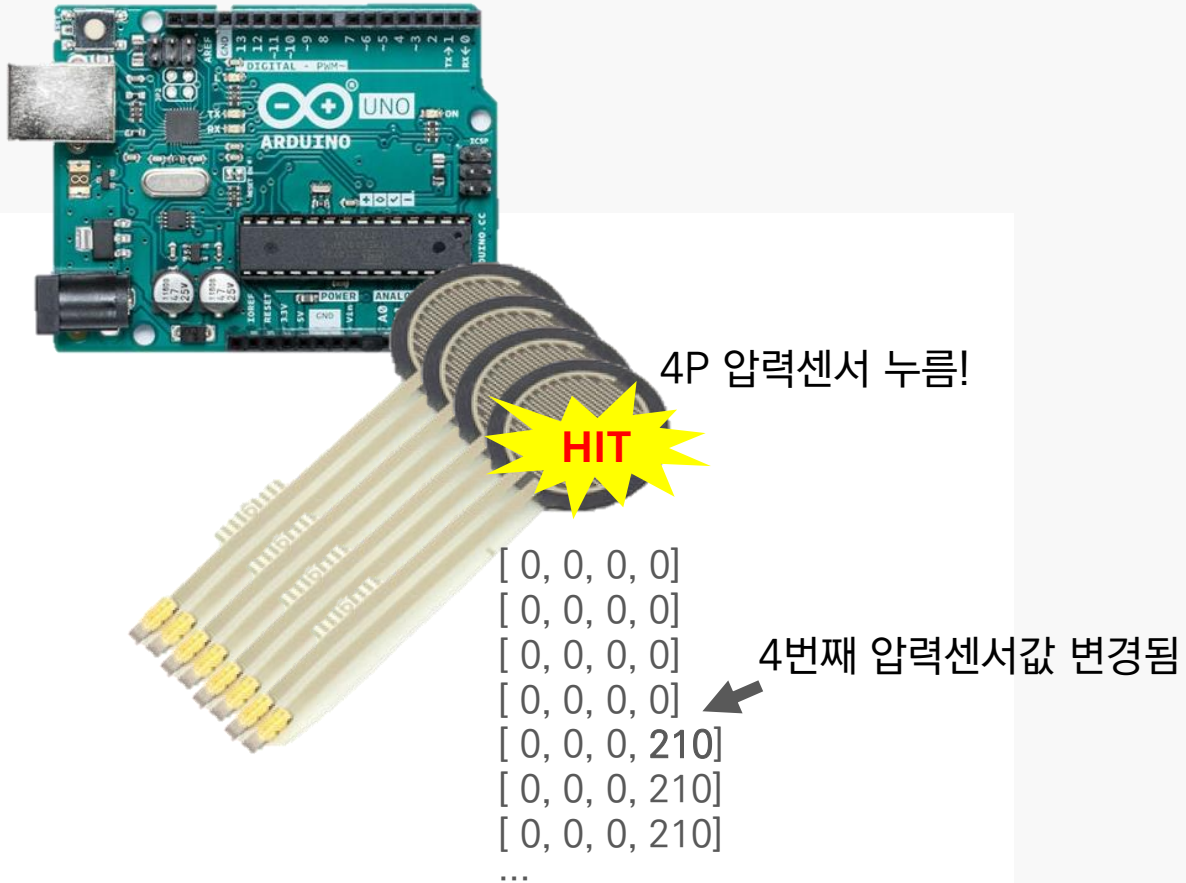


*** 4P mode 와 같은 순서 로직!**

하지만 게임 시작 전 Player 들을 초기화 할 때 2P와 3P는 Dead 상태로 초기화하여 1P와 4P 의 대결로 게임 진행함

03 작품 구현 - 센서 입력

〈아두이노〉



〈라즈베리파이〉

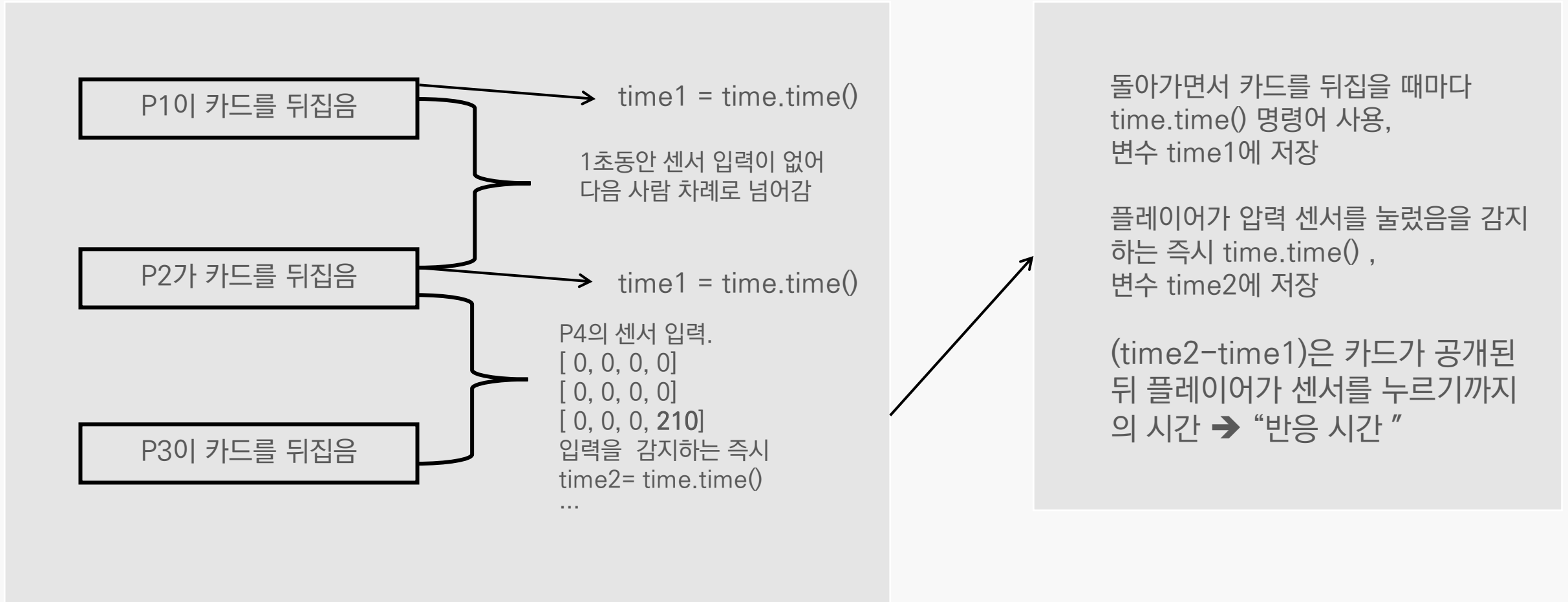


- 카드를 뒤집고 1초 동안 Serial Read
- 플레이어의 센서가 특정 값 이상일 때 종을 쳤다고 판단!

Serial 통신

03 작품 구현 - 반응 속도

◆ 파이썬 time 모듈 사용 → `time.time()` : 컴퓨터의 현재 시각을 구하는 함수



03 작품 구현 - 랭킹

- ◆ rank.txt: 현재까지 진행된 게임들에서 최단 반응시간 TOP10 플레이어들의 이름과 반응 시간을 저장한 파일



P1 0.030
P3 0.120
P4 0.155
...
P1 0.909

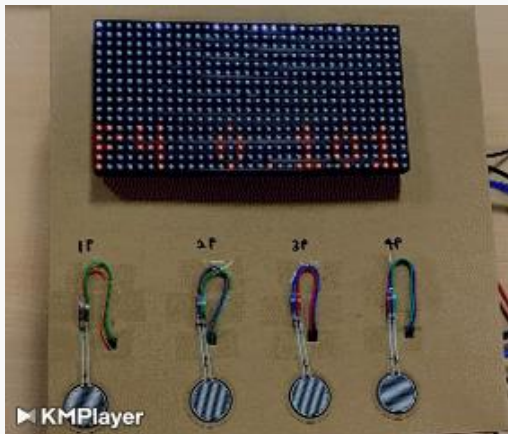
rank.txt

① rank.txt 파일 read



③ 새로운 TOP10 write

② 게임 종료 전, 현재 플레이어들의 최단 반응시간들과 rank.txt의 TOP10 반응시간들을 비교하여 다시 TOP10 랭킹 갱신



④ LED에 TOP10 출력
(스크린 크기의 한계가 있기 때문에 엔딩 크레딧 효과를 줌)

갱신된 TOP10 출력

03 작품 구현 - 역할

곽창민

라즈베리파이와 아두이노 보드 시리얼 통신
아두이노 보드에 압력센서 연결
게임 중 측정된 반응속도를 텍스트파일에 저장
LED Matrix를 통해 엔딩 크레딧 형식으로 텍스트 파일 내용 출력
(공통)게임 로직 테스트 및 디버깅

김재은

Class를 이용한 게임 로직 구축
Pygame 모듈을 사용해서 LED Matrix가 없어도
개발 진행이 수월하도록 함
Github를 통한 진행 상황 제출 관리
(공통) 게임 로직 테스트 및 디버깅

명준모

LED Matrix에 출력 가능하도록 문자 및 숫자 배열 작성
파이썬 게임 소스에서 LED Matrix에 문자, 숫자 출력 함수 작성
LED Matrix를 통해 엔딩 크레딧 형식으로 텍스트 파일 내용 출력
Pygame으로 게임 실행 시, 키보드 입력 구현
(공통)게임 로직 테스트 및 디버깅

04 아쉬운 점 & 학습효과

01 아쉬운 점

- ❖ 게임 시작 전 사용자의 아이디를 직접 입력 받아 사용하지 못한 점
- ❖ LED 밝기 조정을 하지 못한 점
- ❖ 사용자의 아이디와 반응 시간을 저장할 수 있는 게임 서버를 구현하지 못한 점

02 학습 효과

- ❖ 아두이노와 라즈베리 파이의 시리얼 통신
- ❖ 수업시간에 배운 명령어를 라즈베리파이에서 이용해봄으로써 리눅스 환경에 적응할 수 있는 기회가 됨
- ❖ 게임 로직을 구현하며 파이썬을 이용한 개발 경험을 쌓게 됨

데모 동영상 링크

https://www.youtube.com/watch?v=721UDB_9kU&t=4s

- ❖ 게임 시작 전 사용자의 아이디를 직접 입력 받아 사용하지 못한 점
- ❖ LED 밝기 조절을 하지 못한 점
- ❖ 사용자의 아이디와 반응 시간을 저장할 수 있는 게임 서버를 구현하지 못한 점